**Progress Report**

Introduction:

So far, I have focused on training the model, testing its performance, and addressing any challenges that have come up during this process and fortunately, the workflow proceeded smoothly, and the model has been performing well on both the training and testing data.

**Model Building Approach:**

**Naive Bayes Classifier (MultinomialNB):** A simple model for text classification. It was chosen because it's easy to use and works well with text data. **CountVectorizer:** This tool converts the text into numbers so the machine learning model can understand it. **Streamlit:** Used to create an interactive interface where users can input text and get emotion predictions. It's not part of the model itself, but helps with testing and user interaction.

Testing Process:

Train/Test Split: We split the data into training and testing sets. The training set was used to train the model, and the test set helped evaluate its performance. Cross-validation: We used 5-fold cross-validation to check how well the model generalizes to new data and avoid overfitting.

Model Evaluation:

Accuracy: We measured the overall performance of the model using accuracy.

Classification Report: This gave detailed performance metrics for each emotion. Mean Squared Error: We used to measure prediction accuracy, though it's more common in regression tasks.

Challenges and Limitations:

Overfitting: Initially, the model worked well on the training data but struggled with the test data. To address this, I used 5-fold cross-validation, which helped improve the model's performance and reduce overfitting. Data Imbalance: The dataset may have an uneven distribution of emotions, which might cause the model to favor predicting the more common emotions. I haven't used any specific techniques to handle this issue. Model Performance: The Multinomial Naive Bayes classifier worked well for text classification, but its performance could be improved with more advanced techniques

Improvements and Optimizations:

Cross-validation: Used to check the model’s stability and avoid overfitting. Feature Engineering: Cleaned the text data by removing punctuation, numbers, and making everything lowercase. Model Saving: Saved the model with joblib to use it again later for predictions.