Goals

<u>Goal-1:</u>

a. How can you frame this problem as a machine learning problem?

Step	Description
1. Data Preparation	We have labelled data, where each sentence corresponds to an intent (label).
2. Model Selection	The task is a natural language understanding problem, so DL models such as transformers (for example, BERT) are appropriate as they can handle the complexities of text.
3. Feature Engineering	Mainly tokenization, padding, and label encoding.
4. Training	We can train a transformer model to learn the relationship between the tokenized input (sentence) and the output (intent label).
5. Evaluation	The model's performance is evaluated based on its ability to predict the correct intent for unseen sentences, using metrics like accuracy, F1-score, and confusion matrix. For now, let's proceed with the accuracy.

b. **What are the possible pros/cons of the formulations that you considered?**

I choose BERT over DistilBERT because DistilBERT is a smaller, faster, and more resource-efficient version of BERT, retaining about 97% of its performance while requiring 40% fewer parameters. It is ideal for tasks like intent detection, where computational efficiency and quick inference are prioritized without significant loss in accuracy, and considering the data and resources I have (Colab), I would choose it over BERT.

<u>Pros:</u>

State-of-the-art performance: Transformer-based models like DistilBERT capture complex linguistic patterns and have been shown to perform well on NLP tasks such as intent classification.

Pre-trained model: Using a pre-trained model leverages the vast amount of knowledge it has already learned, improving generalization with less training data.

Transfer Learning: Fine-tuning a pre-trained model like DistilBERT specifically for the intent detection task allows the model to better adapt to the domain of the problem.

<u>Cons:</u>

Adaptability to Domain-Specific Applications: DistilBERT's smaller size may limit its ability to generalize across broad domains without significant fine-tuning, especially in niche areas where more recent models have been optimized for specific tasks.

Potential Overfitting on Small Data: Since DistilBERT has fewer parameters, it might overfit faster when fine-tuned on small datasets without sufficient regularization techniques.

2. Goal 2 - Instructions on how we can reproduce your results: Run the notebook and adjust the .csv path accordingly.

3. Goal-3
a. Why do you think your results make sense?

Appropriate Model Choice: DistilBERT has been fine-tuned for tasks like intent detection and has achieved strong results in various NLP benchmarks, making it a suitable choice for this problem.

Data Preprocessing: Proper preprocessing, like tokenization and label encoding, ensures that the input data is in the right format, allowing the model to focus on learning the relationship between the sentences and the intent labels.

Cross-Validation: The model's evaluation on a held-out test set helps ensure that the results are not overfitting and can generalize well to unseen data.

Early Stopping: The use of early stopping prevents the model from overfitting and ensures that we capture the best-performing model on the validation set.

b. How can you improve your model? Why do you think those improvements will actually improve the results?

1. Hyperparameter Tuning - Learning Rate Optimization, Tuning Batch Size: Fine-tuning the learning rate and batch size might improve the model to converge faster and potentially avoid local minima or overfitting, leading to better results.
2. Data Augmentation: For tasks like intent detection, augmenting the data by paraphrasing sentences or introducing noise (e.g., synonyms, word reordering) can help the model generalize better, as the data size is smaller for this case. Augmenting the training data artificially increases the size and variability of the dataset, which can help the model generalize to unseen sentences.

3. Model Architecture Adjustments: Use of larger models (BERT, RoBERTa): You could try using a larger transformer-based model like BERT or RoBERTa, which might capture more complex relationships in the data. Larger models generally perform better on more complex NLP tasks, but they also require more computational resources, so it’s a complete tradeoff.
4. Fine-tuning on domain-specific data: If the training data is domain-specific (e.g., related to an E-commerce data in this case), fine-tuning on domain-specific datasets can improve performance. Domain-specific data allows the model to learn more relevant patterns and can outperform models trained on generic datasets.