Project Overview

The goal of this project is to develop an NLP system that classifies women's fashion reviews into Recommended or Not Recommended Product.. This classification is just the beginning to build a full system that can automatically analyze product reviews to extract valuable insights.

Modeling Approach and Reasoning

Data Preprocessing

- *Text Cleaning:* Removed missing values, Removed duplicates rows, Removed parentheses, Removed numbers, Removed punctuation, Removed extra spaces, Removed special characters, and Removed stopwords.
- *Text Normalization:* Lemmatization.
- *Tokenization:* No tokenization.
- *Vectorization (only for classical models):*
    - Directly applied TF-IDF (*Term Frequency-Inverse Document Frequency*) to transform cleaned text into numerical features. TF-IDF was chosen to emphasize informative and distinguishing words in reviews.

Data Splitting

Splitting Strategy:

- 80% Training Data: Used to train models and optimize parameters.
- 20% Validation Data: Used for hyperparameter tuning and model selection.
- 20% Test Data: Held out for final evaluation to assess generalization.

Splitting was performed using *stratified sampling* to maintain the class balance across all subsets.

Modelling

- Baseline Model: Logistic Regression with TF-IDF features to establish a starting point.
- Advanced Models:
    - *Random Forest:* For interpretable ensemble classification.
    - *XGBoost:* More complex models.
    - *Fine-tuned BERT*: (Bidirectional Encoder Representations from Transformers) to capture contextual nuances in reviews (Basic Fine-Tuning).

Model Selection: The final model (BERT) was selected based on its superior metrics on the validation set.

Result and Analysis

No	Model Name	Train ROC AUC	Val ROC AUC	Train F1-Score	Val F1-Score
1	Logistic Regression	0.9650	0.9270	0.9430	0.9210
2	Random Forest	1.0000	0.8910	1.0000	0.9230
3	XGBoost	0.9820	0.9030	0.9670	0.9280
4	Fine-Tuned BERT	0.9930	0.9460	0.9850	0.9400

- Logistic Regression served as a very good baseline.
- Random Forest and XGBoost are quite overfitting.

- Fine-Tuned BERT excelled due to its ability to handle contextual and semantic nuances, significantly outperforming other models.

Error Analysis

- Our model always correctly predicts text lengths that are over 350 words, but for less than 350 words our model doesn't perform very well.
- Our model misclassifies review texts that are complex, for example: some reviews start by praising the product but end up with returning the product or explaining her/his dissatisfaction. There are many reviews with this kind of style, making our models difficult to understand.

Potential Improvements with Time

Target Variable Fixing

- There are some labels that are not correctly labeled, this needs to be manually fixed by a human to label it properly.

Data Enhancements

- Expand the dataset to include more diverse and representative reviews.

Data Preprocessing

- Check if there are a lot of slangs, maybe this can make our model have a better performance.
- Add tokenization procedure in classical models like word tokenization or sentence tokenization.
- There are a lot of heights mentioned in the review, like 5 '5 feet, 5' 4 feet etc, this can be retained to get more nuanced to the model rather than just remove it.
- Try other vectorizing techniques such as Word2Vec, CBOW, or Skip-gram.

Feature Engineering

- Use other columns like rating, title, age, positive feedback count, division name, department name and class name.

Model Optimization

- Do some hyperparameter tuning.
- Research on domain-specific pretraining transformer models that has been trained on a larger corpus of women's fashion reviews.
- Threshold Optimization.
- Try other fine-tuning strategies, like freeze some layers.

Others

- Improve code quality and organization
- Create a new project to extract product attributes/aspects mentioned in reviews.
- After that, by combining these 2 projects, we can extract insights for what are the most common reasons that a product is recommended or not recommended.
- We can build a streamlit dashboard so that the end-user can see those insights.