INF 2179 – Winter 2025 Classification Challenge

Introduction

The goal of this challenge is to develop predictive machine learning models using women's clothing e-commerce review data. Participants will tackle two classification tasks:

1. **Recommendation Prediction:** Predict whether a user recommends a product (binary classification: recommended = 1, not recommended = 0).
2. **Star Rating Prediction:** Predict the user's given star rating (1 to 5 stars) using the review text and title.

Dataset

Two CSV files derived from the original Women's Clothing E-Commerce Reviews dataset are provided:

- **Training Dataset**: Contains all original features, including *Review Text*, *Title*, *Rating*, *Recommended IND*, *Age*, and *Division Name*, among others.
- **Testing Dataset**: Structured similarly, used for model evaluation.
- More info can be found here

Models must be trained using **only** the provided training dataset and evaluated on the testing dataset.

1 Task 1: Recommendation Prediction (25 points)

You must use the features **except the star ratings** to predict whether a review indicates product recommendation.

Requirements and Evaluation:

- Compute predictions for the *Recommended IND* column (binary: 0 or 1) in the test set.
- Evaluation Metric: Weighted F1-score.
- Full points (25 points) awarded for an F1-score ≥ 0.90. Scores decrease linearly below 0.90, reaching 0 points for F1-score < 0.60.
- The score calculation follows:

$$\text{Score} = \min\left(25, 25 \times \max\left(0, \frac{F1 - 0.60}{0.30}\right)\right)$$

2 Task 2: Star Rating Prediction (25 points)

Participants will develop a multi-class classifier to predict the exact star rating (1 to 5 stars).

Requirements and Evaluation:

- Compute predictions for the *Rating* column in the test set (integer labels 1–5).
- Evaluation Metric: Accuracy.
- Full points (25 points) awarded for accuracy $\geq 60\%$. Scores decrease linearly below 60%, reaching 0 points for accuracy $\leq 40\%$.
- The score calculation follows:

$$\text{Score} = \min\left(25, 25 \times \max\left(0, \frac{\text{Accuracy} - 0.40}{0.20}\right)\right)$$

Submission Guidelines

Submissions on Quercus must include your Notebook file (**without cleaning your outputs!**) and a single PDF document containing:

1. methodologies used., You may try several methods, but you must clearly highlight the best one in your report, as your grade will be based on it.
2. Clearly reported classification results for the best model, including a **classification report** (precision, recall, F1-score, and accuracy) for both tasks. You can generate such report using:
   from sklearn.metrics import classification_report
3. Responses to the following discussion questions.

Discussion Questions:

1. How does your recommendation model perform across the two classes (recommended vs. not recommended)? Discuss precision and recall insights.
2. How does your star-rating model perform across different ratings (1–5)? Identify easier and harder predictions and analyze the reasons.
3. Discuss real-world implications or limitations of these models in a business setting. Provide at least one recommendation for practical implementation.

Evaluation Breakdown

- **Task 1: Recommendation Prediction**: 25 points
- **Task 2: Star Rating Prediction**: 25 points
- **Methodology, Clarity, and Discussion:** 25 points.
- **Code Reproducibility and Clarity in Notebook:** 25 points.

Important Notes

- The work should be reproducible, i.e., using your jupyter notebook, it should be possible and easy to reproduce the same predictions. If this is not the case, the work will not be accepted! To achieve this, don't forget to fix the random state (random_state parameter). I suggest that you try to re-run the entire notebook to ensure the predictions remain identical.
- The submitted notebook must contain all outputs to validate the correctness of your implementation.
- External datasets are not permitted; all training and testing must use the provided datasets.
- Submissions must be individual; collaboration is not allowed.
- Consider using hyperparameter tuning techniques such as **Grid Search** or **Random Search** to optimize your models.

Submission Deadline: March 21, 2025.
For any questions, please consult the discussion board.

Good luck!