INF 2179 – Winter 2025 Classification Challenge

Hamid Parsazadeh

Personal Challenge

The goal of this challenge is to develop predictive machine learning models using women's clothing ecommerce 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.

Preprocessing

In this challenge we have a clothing company's product review dataset which includes numerical, categorical and textual fields. In order to define and fit any prediction model, the dataset requires the following data cleansing and preparations which repeats for both tasks:

Categorical Features Encoding: Three categorical feature columns: "Division Name", "Department Name" and "Class Name" are required to be encoded numerically (by numerical labels) to make these categorical fields compatible with the fitting model and improve the processing memory usage. Moreover, we assign (-1) to unseen categories in the test dataset (the categories that exist in test dataset however, does not exist in the training dataset) as a fallback label to prevent processing error.

Textual Features Transformation: Two textual feature columns: "Title" and "Review Title" should be transformed and combined to include only top 5000 most relevant words in each row to improve the fitting model's efficiency. TF-IDF (Term Frequency-Inverse Document Frequency) method is the implemented technique for this purpose and in fact the previous step (replacing the missing values with empty string) was taken to prevent system error in this technique.

Missing Values in Textual or Categorical Columns: Initial analysis of the dataset shows we have some missing values in different feature columns. The missing values in columns with textual data: "Title" and "Review Title" should be filled with empty string to prevent processing error in TF-IDF Transformation step and also missing values in categorical data gets the label "-1" similar to the unseen categories in test dataset.

Implementing these preprocessing steps, the dataset is ready for fitting on various models to predict "Recommended IND" (Task1) or "Rating" (Task 2)

Task 1: Recommendation Prediction

In this task we need to define and fit a model to predict whether a customer recommend a product (Recommended IND = 1) or not (Recommended IND = 0) based on their review features which are in numerical, textual or categorical data types.
The simplest compatible model for this binary classification task is Logistic Regression which fits very well in this data and generates Weighted F1-score: 0.89. However, the assignment expects 0.90 or more for this task, so Light Gradient Boost model: LightGBM is fit as the predicting model this task.

Note 1. Learning models such as BERT-based or transformers for NLP models would serve the highest accuracy in such these tasks, but their high processing resources demanding went problematic in grid search step (my old desktop could not process it by GPU so I got infinite grid search through CPU), so I could not test their performance in this dataset.

Note 2. The field "Clothing ID" does not include any useful information for this prediction task; therefore, it is dropped out before training the model

Note 3. Since our dataset is imbalanced (82% of the records are "Recommended IND = 1" in train dataset) and it is a binary classification we set the 'is_unbalance' in LightGBM classifier to "True" to give higher weight to underrepresented group (Recommended IND = 0). It helps to reduce bias toward the majority group and improves Recall for the minority group.

Model Fitting

As discussed above, after testing a few models and considering hardware processing limitations, Light Gradient Boost Model (LightGBM) is selected to fit for this binary classification task. Parameter optimization for this model is performed through *GridSearchCV* by the following components:

- num_leaves: [31, 50, 100]
- learning_rate: [0.01, 0.05, 0.1]
- N_estimators: [100, 200, 500]
- Max_depth: [-1, 10, 20]

After performing the grid search the following values are suggested for each of the parameters:

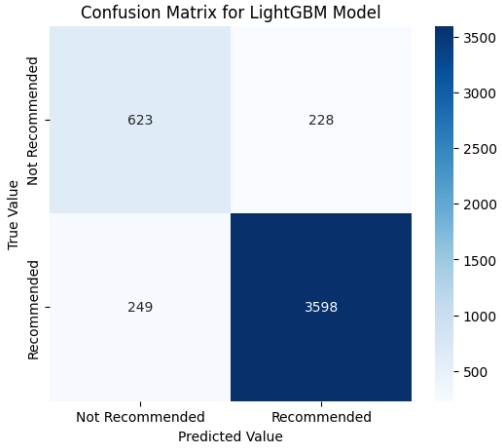
- max_depth: -1
  - Means no depth limit is applied to the decision tree
- learning_rate: 0.05
  - Means that the model updates its weights gradually and does the learning process quite balanced between accuracy and training time.
- n_estimators: 500
  - The number of estimators is in fact the number of boosting iterations (trees) used to train the model. When it is set to 500, it means the model trains 500 trees sequentially, while each one corrects the errors of the previous tree.
- num_leaves: 50
  - Means that each decision tree can have up to 50 leaf nods

Accuracy Check:

After fitting the model, we can measure the best model's accuracy based on Weighted F1-score:

	Precision	Recall	F1-score	Support
Recommended IND = 0	0.71	0.73	0.72	851
Recommended IND = 1	0.94	0.94	0.94	3,847
Weighted Average	0.90	0.90	0.90	4,698

Overall accuracy which is measure by Weighted F1-score is 0.90. Fairly high weighted F1-score for each group (72% & 94%) and similar weighted Precision and weighted Recall rates (90%) means that the model is doing the balanced and reliable job. The heat map of the confusion matrix also shows that the model is doing fairly reliable prediction for both choices of recommending or not recommending a review product.



Task 2: Star Rating Predicting

In this task we need to define and fit a model on the dataset that is already split into train/test datasets to predict the number of stars of a review which can be an integer from 1 to 5 based on the review features which are in mixed types of numerical, textual or categorical data types. This is a Multi-Class Classification task; therefore, potential models would be Random Forest Classifier, BERT-based model or LightGBM.

Although by fitting a Random Forest Classifier model we achieve the overall accuracy of 0.61 which meets the minimum expected accuracy for this assignment, the imbalanced dataset and significantly biased accuracy of individual ratings, make this model not to be a reliable predictor. Therefore, a Light Gradient Boost Model (LightGBM) is fit for this task.

Note 1. Similar to task 1, the field "Clothing ID" is dropped out from the model's features.

Note 2. Since our dataset is imbalanced yet it is a Multi- Class Classification rather than a binary classification we set the 'class_weight' in LightGBM classifier to "balance" to give higher weight to underrepresented groups (rating 1, 2 &3). It helps to reduce bias toward the majority group (rating 5) and improves Recall for the minority group.

Model Fitting

As discussed above, after testing a few models and considering processing limitations, Light Gradient Boost Model (LightGBM) is selected to fit for this Multi-Class Classification task. In order to optimize the processing time, the bin size is set to 63 (rather than 255) and in TF-IDF vectorizer process

the max_feature is set to 1000. Similar to task 1, GridSearchCV is used for Parameter Optimization with the same parameter ranges in task 1 and here is the search result for each parameter:

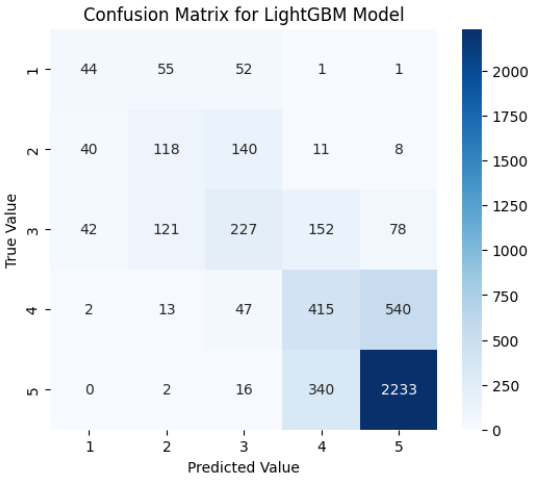
- max_depth: -1
  - Means no depth limit is applied to the decision tree
- learning_rate: 0.05
  - Means that the model updates its weights gradually and does the learning process quite balanced between accuracy and training time.
- n_estimators: 500
  - The number of estimators is in fact the number of boosting iterations (trees) used to train the model. When it is set to 500, it means the model trains 500 trees sequentially, while each one corrects the errors of the previous tree.
- num_leaves: 100
  - Means that each decision tree can have up to 100 leaf nods

Accuracy Check:

After fitting the model, we can measure the best model's accuracy based on Weighted F1-score:

Rating	Precision	Recall	F1-score	Support
1	0.34	0.29	0.31	153
2	0.38	0.37	0.38	317
3	0.47	0.37	0.41	620
4	0.45	0.41	0.43	1017
5	0.78	0.86	0.82	2591
Weighted accuracy			**0.65**	

Overall accuracy is 0.65 which meets the minimum expected accuracy for this task. However, it is not supporting for reliable predictions especially for ratings 1 to 4 which their accuracy is even below 50% Weighted F1-score. This biased prediction is mainly due to imbalanced train data and low number of learning records in those ratings. I tried a few potential solutions as manually calculating the class (ratings) weights or Re-Sampling to increase the number of training records with rating 1 to 4, and did not get any noticeable improvement. Maybe downsizing from rating 5 helps which I did not try.



Discussion Questions:

1. ***How does your recommendation model perform across the two classes (recommended vs. not recommended)?***

Overall, 72% accuracy on "Not Recommended" and 94% accuracy on "Recommended" prediction by this model, means that the model is doing very reliable prediction especially in "Recommended" reviews. The main reason for higher accuracy on recommending products is that almost 82% of the

records in both training and test datasets are recommending the reviewed products. It means more records for training the model and as a result, more accurate prediction.

While the test dataset includes 851 records as “Not Recommended” reviews, the model predicts 623 of them correctly, which means 73% Recall for this group. This rate for “Recommended” records is 94% (3,598 correct predictions out of 3,847 total recommended records in test dataset)

The other accuracy measure, Precision calculates the rate of correct prediction out of total prediction in each group: “Recommended” or “Not Recommended”. While the model has predicted 872 reviews as “Not Recommended”, only 623 of them are correctly “Not Recommended”, which means 71% Precision accuracy for “Not Recommended” group. This rate for the other group is 94%.

2. ***How does your star-rating model perform across different ratings (1–5)? Identify easier and harder predictions and analyze the reasons.***

Overall accuracy of 65% meets the assignment's minimum expectation. However, it is not particularly reliable accuracy, as evidenced by the individual ratings, which show that, with the exception of rating 5, the rest have a significantly lower weighted accuracy. The main cause of this biased accuracy score is the imbalanced trained dataset toward rating 5 and a scarcity of records for training the model for ratings 1, 2, and 3, particularly having just 689 records for training rating 1 does not appear sufficient. Over 10,000 training records with rating 5 account for over 80% accuracy in predicting 5 star ratings, however they are 15 times greater than the amount of training records with rating 1. As a result, the model's prediction is skewed to 5.

3. ***Discuss real-world implications or limitations of these models in a business setting. Provide at least one recommendation for practical implementation.***

In a business setting, the models utilized in Tasks 1 (Customer Recommendation Prediction) and 2 (Star Rating Prediction) have important implications for customer experience management. Task 1 helps firms in identifying customers who are likely to recommend products, which allows for more targeted marketing and service enhancements. However, model limitations, such as biased or imbalanced data, might lower accuracy in forecasting disgruntled customers. To improve classification performance, it is recommended that the model be retrained on new data on a regular basis and sentiment analysis be included.

Task 2 involves predicting star ratings based on reviews, which allows businesses to detect sentiment trends and increase product satisfaction. However, ratings are subjective, and imbalances (for example, an excessive number of 5-star reviews) might bias projections.

To improve this, organizations should use SMOTE (Synthetic Minority Oversampling Technique) to balance the dataset and add sentiment ratings as additional features. To maximize the usefulness of both models, firms should implement real-time monitoring, include forecasts into feedback dashboards, and proactively improve customer service based on model findings.

Also in real world business, we can contact the dataset owner and request to receive further features for the products so we can include them in the model based on the “Clothing_ID”. This ID field does not help individually in the model but it can help to append further product details to the dataset an include the in the model which it can help to increase the prediction’s accuracy in both tasks.