Final Report - Task 5

Task:

This report addresses the task of predicting the beer style based on specific ingredients—such as hops, yeasts, and fermentables—used in the brewing process of beer.

Lead:

The lead contributor for this task was Jordan Van.

Methods:

We experimented with MLPClassifier, Deep Neural Networks (DNNs), Random Forests, XGBoost, and an ensemble combining these approaches. The MLPClassifier from scikit-learn served as our baseline due to its simplicity and its ability to model non-linear patterns. To enhance performance, we also constructed a soft-voting ensemble of three MLPClassifiers with different hidden layer sizes and regularization values. A Deeo Neural Network(DNN) was used to capture deeper detailed patterns through multiple hidden layers. Random Forests were selected for their robustness, interpretability, and their effectiveness in handling large numbers of input features, while XGBoost was chosen for its effectiveness on imbalanced data and its ability to model complex interactions.

Hyperparameters were tuned manually—for example, testing learning rates (0.01–0.0005) for DNNs and adjusting estimator counts and tree depth for tree-based models. Final configurations were selected based on validation accuracy and F1 score improvements over the baseline. Ultimately, we discovered that combining the four models through ensembling using weighted soft voting gave the best performance. By averaging their predicted probabilities, the ensembled model leverage the strengths of all 4 models, achieving the best results on the development set.

DNN: Learning Rate

Learning Rate	Validation Accuracy	Validation F1 score
0.001	0.373	0.273
0.005	0.328	0.205
0.0001	0.368	0.240

Submission Model Details:

Our best-performing model was an ensemble of four classifiers: a Deep Neural Network (DNN), Random Forest, XGBoost, and a soft-voting ensemble of three MLPClassifiers. Each model was trained independently, and their predicted probabilities were combined using weighted soft voting. We assigned weights of (5, 2, 2, 1) to the DNN, Random Forest, XGBoost, and MLP ensemble, respectively. Predictions were made by averaging the weighted outputs and selecting the highest-probability class.

Preprocessing involved converting sparse feature representations to dense matrices using SciPy's coo_matrix, followed by feature standardization via scikit-learn's StandardScaler. The DNN was implemented in PyTorch and wrapped with Skorch. It consisted of three hidden layers with 256 ReLU units each and a softmax output layer. The model was trained using the Adam optimizer with the NLLLoss criterion, a learning rate of 0.001, a batch size of 256, for 18 epochs with data shuffling enabled. The Random Forest used scikit-learn with 100 estimators, bootstrap enabled, OOB scoring enabled, and a random state of 4. XGBoost was implemented using the xgboost library with 400 estimators, a learning rate of 0.1, and a maximum tree depth of 6. Both subsample and colsample_bytree were set to 0.8. The model used the multi:softprob objective and was evaluated using the mlogloss metric. The MLP ensemble included three scikit-learn models, each with four hidden layers and ReLU activation. The architectures were (150, 150, 150, 150), (100, 100, 100, 100), and (50, 50, 50, 50), using alpha values of 0.001, 0.001, and 0.01, respectively. All models were trained with a learning rate of 0.001, a maximum of 500 iterations, early stopping enabled, and distinct random states.

Results:

Performance Summary:

Model	Accuracy	F1	Precision	Recall	Loss
DNN	0.373	0.273	0.299	0.267	2.809
Random Forest	0.427	0.328	0.507	0.278	3.876
XGBoost	0.408	0.306	0.410	0.276	2.128
MLP	0.402	0.285	0.358	0.269	2.208
Ensemble	**0.469**	**0.371**	**0.531**	**0.326**	**1.957**

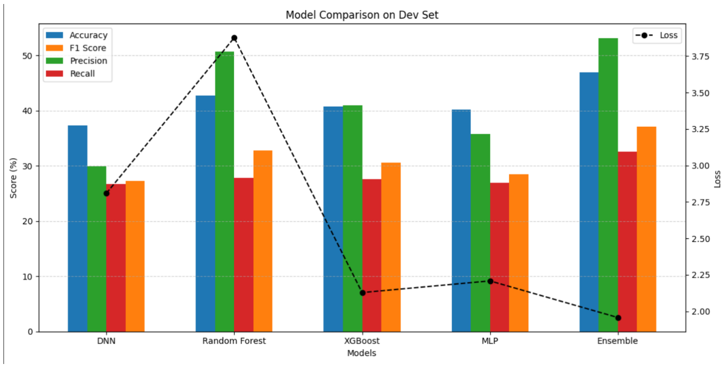
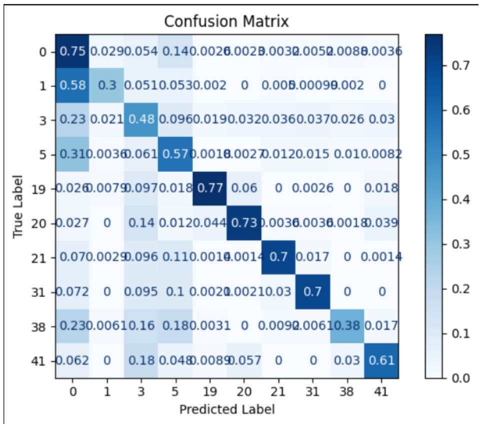


Figure 1: Performance comparison across models.



	0	1	3	5	19	20	21	31	38	41
0	0.75	0.029	0.054	0.14	0.0026	0.0023	0.0032	0.0052	0.0088	0.0036
1	0.58	0.3	0.051	0.053	0.002	0	0.005	0.00099	0.002	0
3	0.23	0.021	0.48	0.096	0.019	0.032	0.036	0.037	0.026	0.03
5	0.31	0.0036	0.061	0.57	0.0018	0.0027	0.012	0.015	0.01	0.0082
19	0.026	0.0079	0.097	0.018	0.77	0.06	0	0.0026	0	0.018
20	0.027	0	0.14	0.012	0.044	0.73	0.0036	0.0036	0.0018	0.039
21	0.07	0.0029	0.096	0.11	0.0014	0.0014	0.7	0.017	0	0.0014
31	0.072	0	0.095	0.1	0.0021	0.0021	0.03	0.7	0	0
38	0.23	0.0061	0.16	0.18	0.0031	0	0.0092	0.0061	0.38	0.017
41	0.062	0	0.18	0.048	0.0089	0.057	0	0	0.03	0.61

Figure 2: Normalized confusion matrix (Top 10 styles).

As shown in both the bar chart (Figure 1) and performance table, the ensemble model outperformed all individual models across nearly every metric. It achieved the highest accuracy (46.9%) and F1 score (0.371), along with the best precision (0.531) and lowest loss (1.957). Compared to the baseline MLPClassifier—which had an accuracy of 40.2% and F1 of 0.285—the ensemble yielded significant improvements, validating our strategy of combining complementary models.

Among individual models, Random Forest had the highest precision (0.507), but suffered from the highest loss, indicating overconfidence in incorrect predictions. XGBoost offered a strong balance of performance and stability. Interestingly, the MLPClassifier slightly outperformed our deep neural network (DNN), likely due to more stable training and fewer tuning dependencies. The DNN underperformed relative to expectations, potentially due to manual hyperparameter tuning and increased model complexity.

Error analysis using the confusion matrix (Figure 2) revealed frequent misclassification between closely related styles of beer. While dominant classes were predicted well (e.g., label 0 with 75% accuracy), less frequent styles showed more dispersion across predictions. This suggests that ingredient overlap, class imbalance, and subtle stylistic differences contribute to classification challenges, particularly for underrepresented styles.