Problem 1:

Bag-of-Words Design Decision Description

Our Bag-of-Words design used the sklearn CountVectorizer class rather than implementing our own to determine our vocabulary set. Specifically, we cleaned the data set by converting each word to lowercase, since we decided that there was likely no correlation between case and complexity. Second, we chose to keep stop words like 'the', 'and', and 'is' because we decided that they would help the model differentiate sentences of different structure (a more difficult text may use more stop words to break up complex sentences). Then, we removed occurrences of words in only 1 document or in more than 90% of documents. This was in part to keep the vocabulary size down (most tests ran around 50k items in vocabulary) and because very rare or very common words have less impact on reading level (super common stop words or made up words likely won't contribute much to a document's rating). Lastly, we used the ngram parameter from CountVectorizer to account for both single words and bigrams that may give away difficulty such as repetitions of pairs of words like 'not only' or 'such that'. We chose to ignore new words encountered in the test for the same reason we ignored words that appeared in only 1 test document. We used a word count instead of binary values because we believe that the number of time a word is seen in a document is correlated to its complexity.

Cross Validation Design Description

We again used sklearn's StratifiedKFold and GridSearchCV to perform cross validation on our train set. We used 5 folds which landed us with 4446 in test and 1111 validation entries per fold. Folds are split at random chosen with shuffled indices in the data set so that there was no bias carried over from the ordering of the train set. We stored the optimal parameters based on each model's performance over AUROC, searching through 7 logarithmically spaced regularization strengths and both L1 and L2 regularization for a total of 14 hyperparameter combinations. AUROC was chosen because it presents a better assessment of classifier performance over other metrics like accuracy. The optimal parameters were assigned to a model using the best_estimator_ attribute.

Hyperparameter Selection for Logistic Regression Classifier

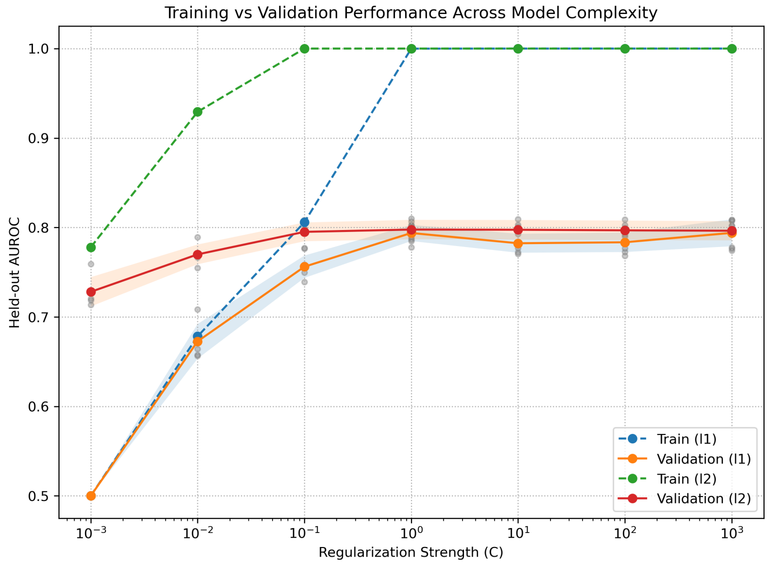
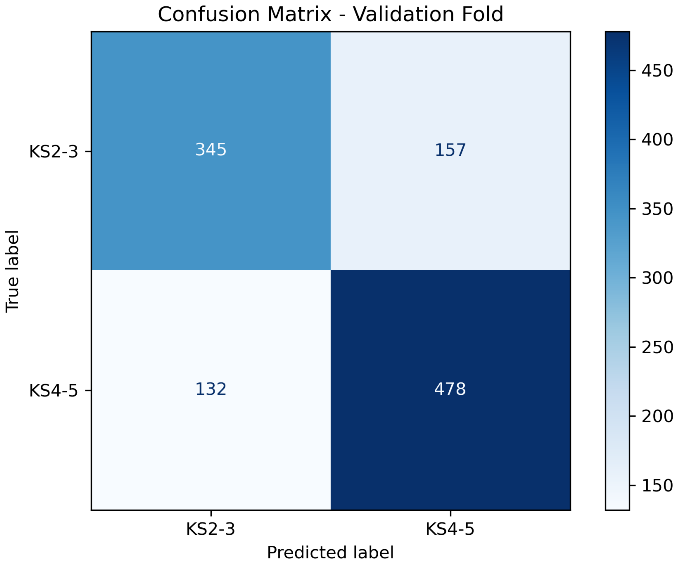


Figure: AUROC on training and validation sets as a function of regularization strength (C). Each point shows average performance across 5 folds, with shaded regions indicating ±1 SD. Low C values (<0.0.1) show underfitting (both scores low), while high C values (>10) show overfitting (train high, validation drops).

Logistic Regression is a strong choice for determining reading level, as it deals well with sparse features and high dimensions that come with the bag of words data representation. We explored both L1 and L2 regularization as well as 7 regularization strengths spaced over the domain 10^{-3} to 10^{3}. The type of regularization affects whether weights will end up as zero (feature gets ignored) or not. We found that L2 performed slightly better, which makes sense, as we wanted words with little weight to still affect the model's prediction and not simply be ignored. For C, a higher value will penalize the weights less than a low value, and it is used to prevent overfitting. The optimal C value fell at 1, where validation AUROC just started to plateau just under 0.8. The model was trained with the liblinear solver from sklearn. No convergence issues were had, as we set the max iterations to 1000. Our final model's parameters were L2 regularization with a C value of 1.0, found via searching over all 14 combinations of C values and both regularization patterns.

Analysis of Predictions for the Best Classifier



Our model tended to over-predict readings as higher level, as shown in the confusion matrix. Looking at the average length of the misclassified texts, they appear to be the same (around 79 words). We did notice, however, that the most misclassified authors wrote lower level texts in much older english, such that the model perceived the old words to mean the text was more difficult than reality. Authors like M. G. Lewis and Robert Louis Stevenson were classified incorrectly often and happened to write around the 18th century. The model tends to perform poorly on documents with a low score but historical language.

Report of Test Set Performance

Our model's ultimate test set performance received an AUROC of 0.689 which is lower than the 0.79 on train data. This likely means that the performance on validation splits overestimated the model's ability to generalize to new data, meaning that the model overfit to the train data. Overall, the search procedure gave a rough but imperfect estimate of model performance.

Problem 2:

Feature Representation

We replaced the Bag-of-Words representation used in Problem 1 with contextual BERT embeddings provided by the assignment dataset. Each passage is represented by a dense vector of 768 dimensions, pre-computed using a transformer model trained on large-scale English corpora. These embeddings encode semantic and syntactic context, allowing the classifier to recognize similar meanings expressed with different wording.

- Training data: x_train_BERT_embeddings.npz → shape (5557, 768)
- Test data: x_test_BERT_embeddings.npz → shape (1197, 768)
- Labels come from y_train.csv, mapping Key Stage 2–3 → 0 and Key Stage 4–5 → 1.

The vectors were loaded with a custom load_arr_from_npz() function that ensures alignment with x_train.csv. No additional normalization or feature concatenation was applied in this baseline, although future extensions could append readability and sentiment metrics from x_train.csv for feature fusion.

Cross Validation Design

To evaluate and select the best classifier configuration, we used a 5-fold `StratifiedGroupKFold` cross-validation procedure with author-level grouping. This ensured that no author's writing style appeared in both the training and validation folds, preventing information leakage between sets. Each fold trained on roughly 80% of the data and validated on the remaining 20%. The evaluation metric was the Area Under the ROC Curve (AUROC), chosen because it is threshold-independent and matches the leaderboard metric.

Hyperparameter tuning was performed using `GridSearchCV`, which exhaustively searched across the specified parameter grid and automatically refit the best-performing model on the full training data. Within each fold, AUROC was computed on the validation subset, and the mean AUROC across all five folds determined the best hyperparameter configuration. All models were trained exclusively on the provided `x_train.csv` and `y_train.csv` data; test data were reserved solely for final leaderboard prediction.

Classifier and Hyperparameter Search

We evaluated three classifiers on the 768-dimensional BERT embeddings: Logistic Regression, Linear SVM, and a Multi-Layer Perceptron (MLP). Each model was tuned using GridSearchCV with 5-fold StratifiedGroupKFold cross-validation and AUROC scoring. The following hyperparameter grids were explored:

- Logistic Regression regularization strength C 0.0001, 0.001, 0.01, 0.1, 1, 3, 10, 30, 50 and L1-L2 penalties
- Linear SVM with Margin regularization C 0.0001, 0.001, 0.01, 0.1, 1, 3, 10, 30, 50
- MLP Neural Network hidden_layer_sizes (64,64), (64,128), (128,64), (128,128), (128,256), (256,256) and alpha 1e-5, 1e-4, 1e-3, 1e-2

Overall, Logistic regression kept showing the best results. To optimize the Logistic Regression classifier, we performed a grid search over the regularization parameter C, which controls model complexity. In LR, small values of C enforce strong regularization, limiting overfitting by constraining the magnitude of learned coefficients, while larger values reduce regularization and allow more flexible decision boundaries. The chosen grid spanned twelve logarithmically spaced values from 1e6 to 100, covering both extreme underfitting (left side) and overfitting (right side) regimes. Each configuration was evaluated using 5-fold StratifiedGroupKFold cross-validation with AUROC as the scoring metric. The grid was designed wide enough to find the U shape that is desired in finding a model that generalizes best. No convergence or step-size issues were observed because the liblinear solver handles small datasets and sparse features efficiently, and the search was limited to a reasonable number of folds and regularization settings.

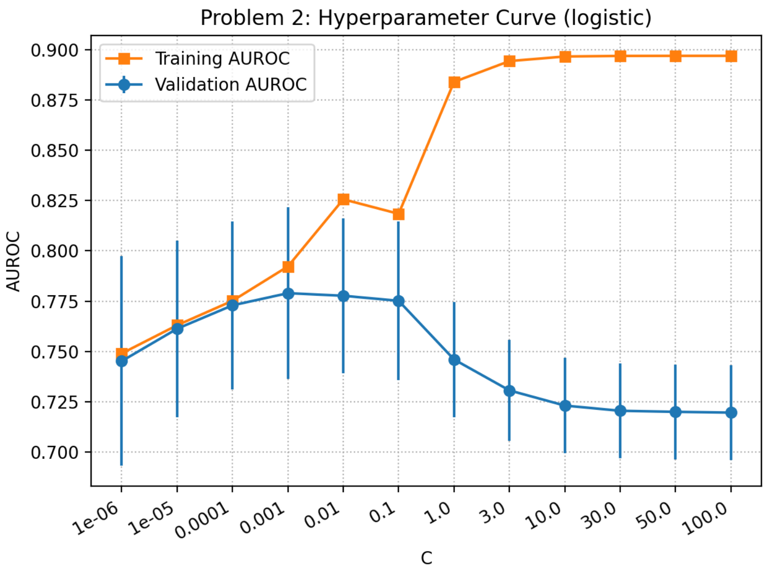
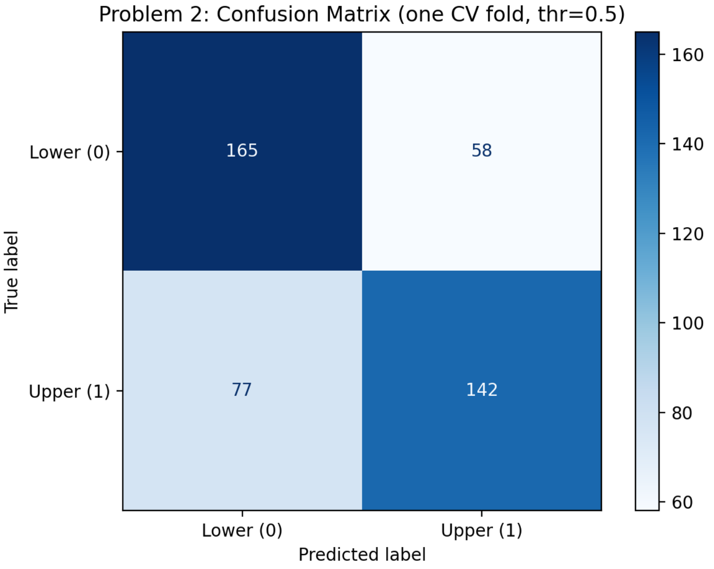


Figure 1: Training (orange) and validation (blue) AUROC as a function of the regularization parameter C.

Validation performance peaks around C = 0.001 and declines for higher values as the model overfits (creating that coveted "U" shape), while training performance continues to rise toward 0.90. Error bars show mean AUROC across folds. For each configuration, the mean validation AUROC was computed and compared across folds. Validation performance improves steadily up to C = 0.001 with L2 penalty, reaching a peak mean AUROC of approximately 0.78, before declining as the model becomes over-regularized or overfits (the coveted "U" shape has been reached). Meanwhile, training AUROC continues to increase beyond this point, indicating the classic overfitting behavior. The difference between training and validation curves widens substantially for C 1, confirming that larger values of C reduce generalization performance.

Error Analysis



True Lower (0): 165 correct, 58 incorrect; True Upper (1): 142 correct, 77 incorrect. Color intensity indicates sample counts. The model performs slightly better on lower-level passages but confuses borderline texts with mixed lexical and syntactic complexity. Group-based folds make this matrix not directly comparable to the Problem 1 results. Figure 2 presents the confusion matrix for the best classifier selected in Section 2C i.e. Logistic Regression trained on 768-dimensional BERT embeddings. Evaluation was performed on one held-out validation fold (unseen during training). The model correctly identified 165 lower-level and 142 upper-level passages, achieving an overall accuracy of roughly 70%. False positives (58 cases) arise when the model predicts "upper-level" for passages that are simpler. These are often texts containing rare or abstract vocabulary within otherwise straightforward syntax. False negatives (77 cases) occur primarily for short but semantically dense or metaphorical passages that the model underestimates in difficulty.

Relative to the Bag-of-Words classifier in Problem 1, this BERT-based model makes fewer purely lexical mistakes but still struggles with stylistic variation among authors. Because Problem 2 uses author-grouped folds (StratifiedGroupKFold) and dense contextual embeddings, its confusion matrix is not directly comparable to Problem 1's, which was computed under random stratification using sparse BoW features. Nonetheless, the distribution of errors shows improved semantic discrimination and reduced over-reliance on word frequency patterns.

Report of Test Set Performance

The final BERT-based classifier achieved a cross-validated AUROC of 0.78 and a leaderboard test-set AUROC of 0.73. Compared with the Bag-of-Words model from Problem 1 (CV 0.80, test 0.69), the BERT model provided more stable generalization across authors, indicating that contextualized semantic features capture reading difficulty more robustly than raw lexical frequencies. Future improvements could include integrating readability metrics or fine-tuning BERT on educational corpora to better handle historical or archaic language that still confuses the classifier. Overall, the BERT pipeline met expectations for Problem 2, balancing accuracy and generalization while demonstrating clear advancement beyond the baseline.