Project Report

Data Mining and Machine Learning

Group 42
Zarja Javh Dobernik; s1169080
Lan Stare; s1169977

January 7, 2026

1 Preliminaries (600 words max)

The goal of this project was to design a classification system that performs well in terms of AUC on an unlabeled test set. The training data consists of 4000 labeled examples evenly split between two classes, each described by 64 numerical attributes. The test set contains 50000 unlabeled examples with known class priors of 0.4 for class 1 and 0.6 for class 2.

As an initial step, we examined the data for missing values and anomalies. No missing values were detected, and all attributes were numerical and already scaled well, making extensive preprocessing unnecessary. Standardization was still applied for models sensitive to feature scale; logistic regression, neural networks, and support vector machines.

We evaluated five different classification approaches: logistic regression, decision tree classifier, random forest classifier, neural networks, and support vector machines (SVM). Each model was evaluated using stratified 5-fold cross-validation on the balanced training data, with AUC as the primary metric. We also used additional performance measures such as accuracy, precision, recall, and F1-score because we wanted to gain further insight into model behavior mostly to detect possible overfitting.

Logistic regression served us as a linear baseline. It achieved a mean cross-validated AUC of approximately 0.875, with moderate variance across folds. While its performance was stable and interpretable, it was outperformed by all the other classification techniques used which are more flexible non-linear models, indicating that the class boundary is likely non-linear.

Decision trees achieved a mean AUC of roughly 0.891. However, performance varied across folds, and trees showed sensitivity to training splits, which suggests that they aren't robust. While decision trees are easy to interpret, they couldn't compare to ensemble methods.

Random forests delivered a substantial performance improvement, achieving a mean AUC of approximately 0.967 with low variance. The ensemble approach reduced overfitting and more effectively captured complex feature interactions. Training AUC reached 1.0, which raised mild concerns regarding overfitting. To confirm that the performance was not due to chance, we used permutation testing.

Neural networks also performed strongly, reaching a mean AUC of around 0.958. Despite this strong performance, they required more careful tuning and were less transparent in terms of decision-making, which complicated validation and interpretation. They were also more computationally expensive.

Support vector machines with an RBF kernel achieved the highest and most consistent performance. With optimized hyperparameters, the SVM obtained a mean cross-validated AUC of approximately 0.978. Although we could observe a small train–validation gap, permutation tests indicated that the learned structure was meaningful (not due to overfitting).

Based on the performance on these classification techniques, the SVM was selected as the final model due to its superior AUC performance, stability across folds, and robustness.

2 Classification Approach (600 words max)

As mentioned in the previous section, the final classification model selected for this project is a support vector machine (SVM) with a radial basis function (RBF) kernel. We decided on SVM due to its performance and its suitability for high-dimensional numerical data with potentially non-linear class boundaries.

All features are numerical and no missing values were present in the dataset, but we still used feature scaling since it's essential for SVM as the kernel function is sensitive to the magnitude of the inputs. Therefore, all attributes were standardized to zero mean and unit variance using a `StandardScaler`. To ensure correct preprocessing, scaling was embedded within a `Pipeline`, so that the scaler was fitted only on training folds during cross-validation and then applied to validation data.

The SVM we implemented uses an RBF kernel, which introduces two key hyperparameters: the regularization parameter C, controlling the trade-off between margin maximization and classification error, and the kernel parameter y, determining the width of the Gaussian kernel. Since only these two parameters directly affect the model's capacity, hyperparameter tuning was restricted to them. We performed a grid search over a small set of values for C and y, using stratified 5-fold cross-validation and the area under the ROC curve (AUC) as the optimization criteria.

To obtain an unbiased estimate of performance during tuning, the labeled dataset was first split into a training set (70%) and a validation set (30%), while preserving class balance through stratification. The grid search was executed only on the training data. We found out that the best performing parameters were: C=10 with y=scale, achieving a mean cross-validated AUC of approximately 0.98. The resulting classifier showed stable performance across folds, indicating robustness.

The final SVM was configured to output posterior class probabilities. These were used both for ROC/AUC evaluation and for generating the required submission csv file. Since the evaluation metric is AUC, which depends solely on the ranking of samples, we didn not apply any hard classification threshold. Instead, the predicted probability of belonging to class 2 was used as a continuous score to rank all test instances.

After selecting the hyperparameters, the optimized SVM pipeline was retrained on the complete labeled dataset (to maximize the use of available information). The trained model was then applied to the unlabeled test set to produce one score per sample. These scores were written into a CSV file in the same order as the test data (as instructed). Lower values are corresponding to a higher likelihood of belonging to class 1 and higher values to class 2.

To assess model robustness and detect potential overfitting, we performed an overfitting check. We examined the difference between training and validation AUC, which revealed a small but acceptable gap, which is consistent with mild model complexity rather than severe overfitting. Furthermore, we also randomly shuffled class labels and re-evaluated performance. As expected, this resulted in AUC values close to 0.5, confirming that the model's strong performance on the original data reflects genuine structure rather than chance correlations.

Overall, the SVM-based approach provides a strong balance between flexibility, robustness, and ranking performance, making it well suited for this classification task. The only drawback is time complexity so if we needed to examine more instances, we'd quite quickly run into problems with time. Although for the given dataset, the python file was executed and the csv produced in approximately 50 seconds, which is still feasible.

3 Performance Estimation (600 words max)

The primary performance measure is AUC. Since the test set is unlabeled, estimating the expected test performance must rely entirely on the labeled training data. For this reason, we used multiple complementary evaluation to obtain a reliable performance estimate.

The main estimate was derived using stratified 5-fold cross-validation on the training set, with AUC as the evaluation metric. Stratification made sure that each fold preserved the original class balance, thereby preventing biased performance estimates. For the selected support vector machine model, the cross-validated AUC values were consistently high across all folds, resulting in a mean AUC of 0.9776±0.0044.

This value serves as the primary estimate of expected test performance. To further assess generalization ability, we used a separate hold-out validation set using 30% of the labeled data. The classifier was trained on the remaining 70% and evaluated on this validation set. The resulting validation AUC was approximately 0.983, which closely matches the cross-validation estimate. This consistency indicates that the model's performance is stable and not overly dependent on a single data split.

Overfitting was evaluated by comparing training and validation AUC values. The model achieved a near-perfect training AUC, while the validation AUC remained slightly lower, producing a small performance gap of approximately 0.017. Given the complexity of the SVM with an RBF kernel and the dimensionality of the data, this gap does not indicate severe overfitting.

As mentioned in the previous section, we also did a permutation test where the class labels were randomly shuffled and the same cross-validation procedure was applied. The resulting AUC values were centered around 0.5, as expected for random labels. This further confirms that the high AUC achieved by the model on the original data was meaningful.

In conclusion, based on cross-validation, hold-out validation, and permutation testing, the expected AUC on the test set is estimated to lie close to 0.98, with 0.9776 as the central estimate.

4 Other Matters (250 words max)

Since both random forests and neural networks showed promising results, we could use them instead of SVM in some cases. In particular, if complexity became a problem, we would prefer the random forest classifier, since the python file containing random forest code executed in approximately 14 seconds.