Assessing the Impact of Data Corruption and Cleaning on SVM and Neural Network Classification

Adina Nadeem[1], **Syed Ayaan Danish**[1], **Prithvi Rajan Ramamurthy**[1]

[1]Department of Mathematics and Informatics, FU Berlin

***For correspondence:**
adina.nadeem@fu-berlin.de;
rajar99@zedat.fu-berlin.de;
ayaan.danish@fu-berlin.de

[†]These authors contributed equally to this work

Abstract

Goal of the project: To assess how data corruption and cleaning methods affect the performance and feature selection of machine learning models on metabolomic data.
Main results of the project: The results show that both classifiers experienced performance drops in the presence of data corruption, especially when values were missing. KNN imputation was the most effective cleaning method, helping to recover both accuracy and AUC, while mean imputation offered moderate improvement. Row removal was ineffective, as it left the dataset empty due to its high dimensionality. We also observed that feature importance varied significantly across different data conditions, highlighting the sensitivity of biomarker selection to data quality and preprocessing choices.
Personal key learnings: Data quality significantly impacts model performance and feature selection, and different cleaning methods can lead to varying results in both accuracy and biomarker identification.
Estimated working hours: Approximately 10 hours per person
Project evaluation: 2
Number of Words: 1432

Goal

The goal of this project is to assess how data corruption and cleaning strategies affect the performance and interpretability of machine learning models applied to metabolomic data. By introducing controlled missingness and noise into a real dataset, and applying various cleaning methods, we evaluate the impact of preprocessing on classification accuracy and feature selection. This work highlights the importance of data quality in biomarker discovery and shows that preprocessing often influences results more than the choice of machine learning algorithm.

Data and Preprocessing

Data Corruption

In this project, we elected to use the same dataset as in the previous project, namely the MTBLS92 dataset [1]. As mentioned, the goal of this study was to assess the impact of corrupt or missing data. To that end, we created multiple different corrupt versions of the MTBLS92 dataset for comparison. The following versions of the dataset were analysed:

- The original preprocessed dataset
- Two versions with removed values of:
  - 10% Strength

  - 50% Strength
- Two versions with changed values of:
  - 10% Strength
  - 50% Strength
- Six subsequently cleaned versions of the missing value datasets, each cleaned using:
  - Mean Value Imputation
  - KNN Imputation
  - Removal of rows with a missing value

This scheme resulted in a total of 11 different versions of the dataset to be classified. An overview of the datasets used is illustrated in Figure 1.

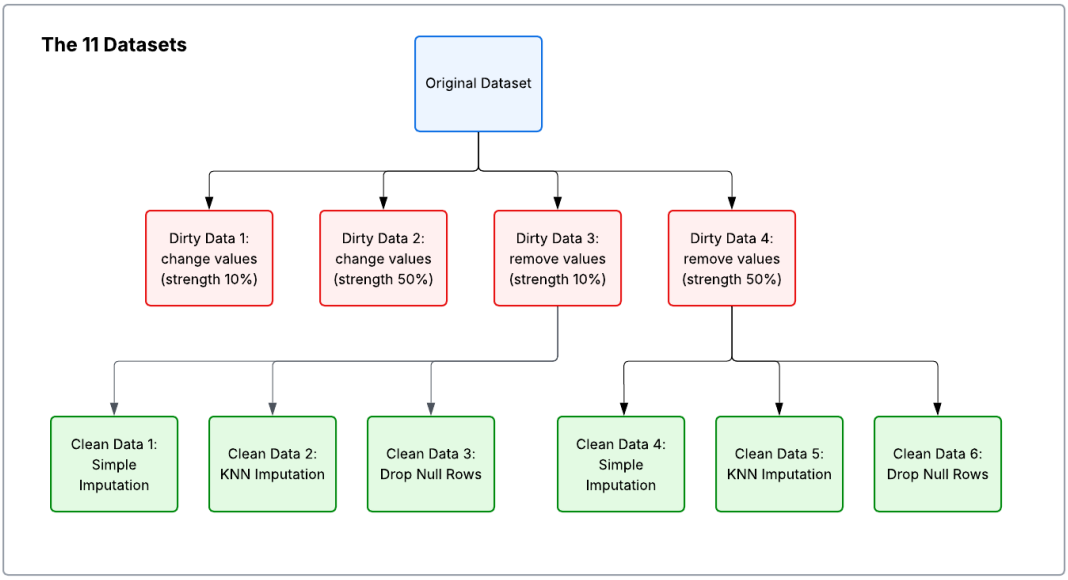

Figure 1. Overview of dataset versions used for the analysis.

To change the dataset values, we defined a function to randomly modify values within the dataset. The randomization is performed by selecting random rows in the numeric columns and replacing their values with new random values drawn from the same column's range. This ensures that the modified values are not unrealistic while also simulating noise in the data. The strength levels of 10% and 50% determined how many rows of the dataset were affected by the change.

Similarly, to remove dataset values, we defined a function to randomly set values within the dataset to NaN (null). The function ensures that missing values are randomly distributed across all eligible rows and columns. The strength levels of 10% and 50% similarly determine the number of rows affected.

Data Cleaning

After introducing missing values into the dataset using random deletion of 10% and 50% of non-class feature values, we applied three different strategies to clean the data:

1. **Mean Imputation:** Missing values were replaced with the mean of each feature using `SimpleImputer` from scikit-learn. This method assumes missingness is random and that the mean is a reasonable estimate for imputation.
2. **K-Nearest Neighbors (KNN) Imputation:** We used KNNImputer from scikit-learn with 5 neighbors to fill missing values based on similarity to other samples. This method takes into

account the structure of the data, making it more adaptive than simple averaging. We did not standardize the features before applying KNN, which might have affected the distance calculations slightly, since features in metabolomics data can vary in scale. However, this was kept consistent across datasets, so comparisons between models remain valid.

3. **Row Removal:** Finally, we experimented with simply removing any rows containing missing values using `dropna()`. However, this method proved problematic: due to the high dimensionality of the data, low number of samples, and the random distribution of missing values, all rows were affected, resulting in empty datasets even with a strength level of 10%. This demonstrated the impracticality of row deletion in sparse or high-dimensional datasets.

These cleaned datasets were then used alongside their unclean counterparts as well as the original dataset to train and evaluate the models, allowing us to assess how different imputation strategies affected performance.

Data Analysis

Methodology

The data analysis methodology remains largely the same as the previous project. The key difference here is that instead of seven different classifiers, the two best performing classifiers from last time were used, i.e, Support Vector Machine and Neural Network. The remaining steps in the analysis methodology, i.e, the feature categorization, encoding, scaling, pipeline assembly, cross-validation, and hyperparameter optimization remain unchanged.

Evaluation Metrics

The key metrics used to evaluate the models were once again precision, recall, F1 Score, and average 5-fold cross validation accuracy. Confusion matrices and ROC curves were also plotted for each model.

Feature Importance Estimation

To identify the most influential features contributing to model predictions, we performed a permutation importance analysis on the best-performing Support Vector Classifier (SVC) and Neural Network models. Unlike tree-based models, SVCs and neural networks in scikit-learn do not expose feature importances by default, making permutation importance a suitable model-agnostic alternative.

After training, we extracted the fitted pipelines and applied `permutation_importance` using 10 random shuffles to evaluate the sensitivity of the model performance to individual feature perturbations. The feature names were reconstructed from the pipeline, combining both the original numeric features and the one-hot encoded categorical features. For each model, the five most important features were selected based on their average decrease in accuracy across permutations and visualized using horizontal bar plots. This allowed us to interpret which metabolites were most critical for classification performance despite the complexity of the models.

Results & Discussion

The performance of both classifiers deteriorated under data corruption and improved after cleaning, though not uniformly. As shown in the AUC trend plot (Figure 2), the highest scores were achieved on the original dataset, with AUCs of 0.789 for the Support Vector Machine (SVM) and 0.800 for the Neural Network (NN). The corresponding accuracies were 0.734 (SVM) and 0.704 (NN), providing a strong baseline for comparison.

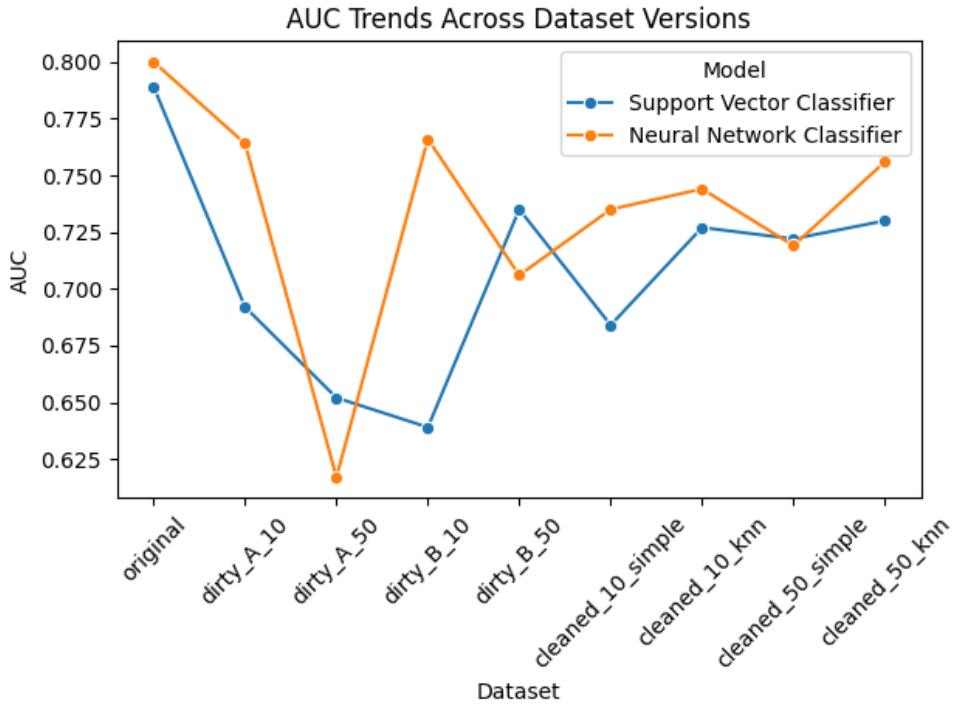

Figure 2. AUC Trends Across Dataset Versions for SVM and Neural Network classifiers.

Impact of Missing Values

Introducing missing values through random deletion (dirty_A) significantly reduced performance, especially for SVM. At 10% missingness, SVM accuracy dropped to 0.562 and AUC to 0.692. In contrast, the NN model was more resilient, with an accuracy of 0.639 and AUC of 0.764. At 50% missingness, both models suffered further, particularly in AUC. These results suggest that missing data degrades not only prediction accuracy but also the model's ability to rank samples effectively.

Impact of Value Corruption

In contrast, corrupting values (dirty_B) was less harmful overall. At 10% corruption, SVM showed high accuracy (0.775), though with a low AUC (0.639), indicating poor class ranking. NN maintained balanced performance, with 0.674 accuracy and 0.766 AUC. At 50% corruption, both models still performed reasonably, suggesting that models tolerate noise better than missing values, especially when the corrupted values remain within plausible ranges.

Effectiveness of Cleaning Methods

Cleaning was applied only to datasets with missing values. Among the three methods, KNN imputation consistently outperformed simple mean imputation in both accuracy and AUC. For example, on the cleaned_50_knn dataset, SVM achieved an AUC of 0.730 and NN reached 0.756—substantial recoveries from the dirty_A_50 condition. Accuracy also improved, though less dramatically. Mean imputation also improved results, but to a lesser extent. Row removal, however, failed entirely: both 10% and 50% dropna datasets were empty due to the high dimensionality and random spread of missing values. This confirms that simple deletion is impractical in high-dimensional biomedical datasets.

Cleaning vs. Corruption

Notably, cleaning had a positive effect only when data was missing. No cleaning was needed for the corrupted datasets, yet those models performed relatively well—especially at 10% corruption.

This emphasizes that missing data disrupts training more severely than noisy data. It also highlights the importance of choosing preprocessing strategies that align with the type of data degradation present.

Changes in Feature Importance

The top five most important features changed across datasets. For example, M59 appeared prominently in the original SVM model but disappeared in most corrupted and cleaned versions. In contrast, features like M27 and M62 were consistently selected, suggesting some degree of robustness. Figure 3 illustrates the overlap of top five features between the original and all other datasets. Interestingly, some cleaned datasets showed *less* overlap than the corrupted ones, indicating that even well-intentioned cleaning can distort model interpretation.

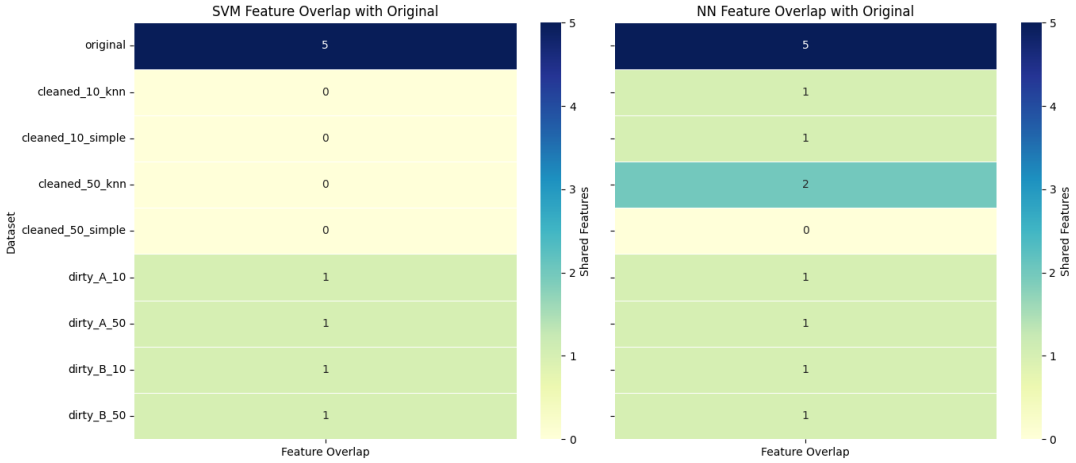

Figure 3. Heatmaps showing the number of overlapping top 5 features between the original model and each corrupted or cleaned dataset, for both the Support Vector Classifier (left) and Neural Network (right). The original models share all 5 features with themselves, while overlap with other datasets varies significantly.

Conclusion

These results highlight the importance of proper data cleaning in maintaining both model performance and biological insight. While machine learning models showed some robustness to random noise, missing values had a far more disruptive effect. KNN imputation provided the most consistent improvements, particularly in AUC. Finally, this study reinforces the idea that evaluating both accuracy and AUC is essential—high accuracy alone can mask poor class separation. Data preprocessing is not just a technical detail; it has a direct impact on the reliability of biomarker discovery.

Contributions

- Adina Nadeem: implemented the classification pipelines and evaluated model performance.
- Syed Ayaan Danish: conducted the data dirtying, imputation experiments and model implementation.
- Prithvi Rajan Ramamurthy: carried out model implementation, feature importance analysis and visualizations.

References

[1] Hilvo, Mika et al. “Monounsaturated fatty acids in serum triacylglycerols are associated with response to neoadjuvant chemotherapy in breast cancer patients.” International journal of cancer vol. 134,7 (2014): 1725-33. doi:10.1002/ijc.28491