E4579 ML in Practice Final Report

Team Member: Iris Shen (hs3469), Viona Xu (zx2472)

In addressing the critical challenge of handling missing data in machine learning, we choose the paper Missing Data Imputation for Supervised Learning[1] (GitHub Repo[2]) to explore the insights in improving model performance through various imputation methods.

Paper Summary

The primary goal of the study in this paper is to assess the out-of-sample performance of machine learning classifiers—specifically decision trees, random forests, and neural networks—when trained on datasets with missing values under different imputation methods (including non-imputed, i.e., one-hot encoded).The paper highlights the impact of missing data on the accuracy of predictive models and proposes that imputation might improve model performance, especially when combined with intentional missing-data perturbation (i.e., artificially introduced missing data).

Imputation Methods Introduced by the Paper

We used 6 different imputation methods.

1. Random Replacement: Missing values are randomly replaced by a non-missing observation.
2. Summary Statistics: Replace the missing data with the mean, median, or mode (used) of the feature vector.
3. One-hot Encoding: Create a binary variable to indicate whether a specific feature is missing or not.
4. KNN: Compute the K-Nearest Neighbors of the observation with missing data and assign the mean of the K-neighbors to the missing data.
5. Prediction Model: Train a prediction model to predict the missing value. In this project, we use random forest
6. Factor Analysis: Perform factor analysis on the design matrix, project the design matrix onto the first N eigenvectors and replace the missing values by the values that might be given by the projected design matrix. In this project, we used principal component analysis (PCA).

Dataset Used

The paper used two benchmark datasets from the UCI Machine Learning Repository: the Adult[3] dataset and Congressional Voting Records[4] (CVRs) dataset. Raw data files can be found on the author's GitHub as well.

For the Adult Dataset (N=48,842), the task is to predict whether the income of one adult will exceed $50K or not based on 6 continuous features like age and 8 categorical features like marital status and race.

For the CVRs dataset (N=435), the task is to classify the party affiliation based on roll call votes of "yes," "no," or missing. As the missing roll call votes often represent the awkwardness of agree or disagree and have strong relationship with their affiliation, the features are stronger correlated compared to the Adult Dataset.

Replication Result

For both dataset, we trained Decision Tree Model, Random Forest Model, and Neural Network Model and we compared the accuracy of different imputation methods and the raw data.

---

[1] Poulos, J., & Valle, R. (2018). Missing Data Imputation for Supervised Learning. Applied Artificial Intelligence, 32(2), 186–196. https://doi.org/10.1080/08839514.2018.1448143 ;

[2] https://github.com/rafaelvalle/MDI/tree/master

[3] Becker, B., & Kohavi, R. (1996). Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20.

[4] Congressional Voting Records. (1987). UCI Machine Learning Repository. https://doi.org/10.24432/C5C01P.

For Adult Dataset, compared with raw dataset, some of the imputation methods can improve the prediction accuracy while some have little improvement. Our main insight is that for a dataset with almost randomly missing values, the selection of suitable models outweighs the use of imputation methods.

Decision Tree

MDI Method	Accuracy
None	81.49%
Random Replace	80.9%
Summary	81.15%
One-Hot	81.52%
Prediction	81.16%
KNN	81.52%
Factor Analysis	81.29%

Random Forest

MDI Method	Accuracy
None	85.3%
Random Replace	85.3%
Summary	85.27%
One-Hot	84.99%
Prediction	85.09%
KNN	81.32%
Factor Analysis	85.55%

Neural Network

MDI Method	Accuracy
None	84.05%
Random Replace	81.45%
Summary	76.64%
Prediction	83.03%
KNN	84.05%
Factor Analysis	84.32%

Adult Dataset Prediction Accuracy

For the CVRs Dataset, post-imputation accuracy was similar to or slightly improved compared to the raw data, likely due to the small sample with highly correlated features and the already high baseline performance of the three classifiers. This suggests that while imputation techniques can enhance data completeness, their impact on predictive accuracy might be limited when the original dataset already supports effective model training.

Decision Tree

MDI Method	Accuracy
None	94.25%
Random Replace	91.95%
Summary	94.25%
One-Hot	94.25%
Prediction	94.25%
KNN	94.25%
Factor Analysis	94.25%

Random Forest

MDI Method	Accuracy
None	96.55%
Random Replace	97.70%
Summary	97.70%
One-Hot	96.55%
Prediction	96.55%
KNN	96.55%
Factor Analysis	96.55%

Neural Network

MDI Method	Accuracy
None	96.55%
Random Replace	96.55%
Summary	94.25%
Prediction	96.55%
KNN	96.55%
Factor Analysis	94.25%

CVRs Dataset Prediction Accuracy

Extension

We implemented and tested a KNN classifier to evaluate its performance in comparison to other models. The below charts show the prediction accuracy for the KNN classifier.

Adult Dataset

MDI Method	Accuracy
None	82.33%
Random Replace	82.46%
Summary	82.5%
One-Hot	77.92%
Prediction	82.22%
KNN	82.33%
Factor Analysis	82.39%

CVR Dataset

MDI Method	Accuracy
None	89.66%
Random Replace	91.95%
Summary	90.80%
One-Hot	91.95%
Prediction	91.95%
KNN	89.66%
Factor Analysis	90.80%

KNN Classifier Prediction Accuracy on Both Datasets

Blockers

The project's code repository was poorly organized, lacking intuitive structure and documentation. Meanwhile, all code was written in Python 2, which is no longer supported, so a complete translation to Python 3 was needed to ensure compatibility and further extension. The code also relies on several discontinued packages, (e.g. *Lasagne* used for neural network training), making replication difficult due to missing training parameters.