|  |
| --- |
| Machine Learning Classification & Missing Value Estimation |

**Mark Lynch**

Georgia State University

*mlynch7@student.gsu.edu*

**Seth Tomy**

Georgia State University

[*stomy1@student.gsu.edu*](http://stomy1@student.gsu.edu)

**Abstract**

In this paper we describe our methodology in Classification of 5 datasets and Missing Value Estimates of 3 datasets.

**1. PROJECT STRUCTURE**

**1.1 Python**

Python was chosen because of its ease of use, it being our primary language, and wide availability of data(pandas, numpy) and machine learning packages.

**1.1.2 Pandas**

Pandas is a powerful tool for working with heterogeneous data and, in conjunction with jupyter notebook, provides a quick means of exploring that data. I serves the data in numpy arrays which are needed to feed to the models in sklearn.

**1.4 SciKit-Learn**

sklearn is a python machine learning framework. We used it extensively to provide us models, cross-validation, grid-searching of hyperparameters

**2. Data**

The datasets varied in number of samples, features, classes and number of missing values.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Number of Samples** | **Number of Features** | **Number of Classes** |
| 1 | 150 | 3312 | 5 |
| 2 | 100 | 9182 | 11 |
| 3 | 6300 | 13 | 9 |
| 4 | 2547 | 112 | 9 |
| 5 | 1119 | 11 | 6 |

**3. MISSING VALUE ESTIMATION**

**3.1 Setup**

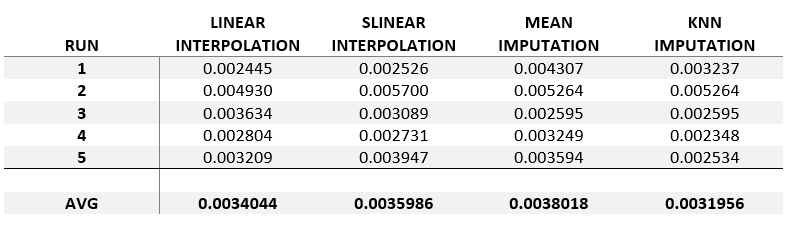
In order to determine the efficacy of our missing value estimation methods, we began by removing 100 known values from each of the three missing-value datasets. This was in addition to the already missing values. All missing and removed values were then set as NaN.

We ran each dataset through two interpolation strategies, linear and slinear, as well as two imputation strategies, mean imputation and knn imputation. The weighted knn imputation method was custom-coded, using numpy and pandas, to find 4 nearest neighbors using euclidean distance, and weight their value contributions accordingly.

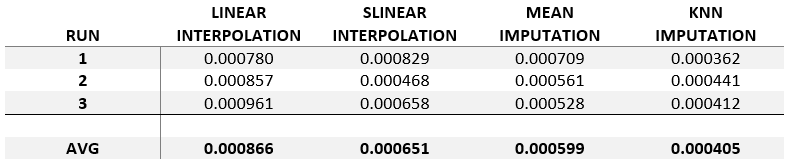
The results of each method were compared to the original dataset, with the initial unknown values replaced by predicted values to reduce error, and compared using sklearn’s mean squared error method.

**3.2 Results**

Mean Squared Error for MissingData1.txt:



Mean Squared Error for MissingData2.txt:



**3.3 Conclusion**

For dataset 1, our weighted knn imputation method performed marginally better than other methods, but in dataset 2 clearly separated itself from the other methods. Our belief is that this was due to the second dataset being substantially larger, allowing for increased similarity with close neighbors. Thus, our weighted knn imputer was utilized for our final estimations. Due to optimization, robustness, and speed issues, we opted, however, to use mean imputation for our classifications.

**4. CLASSIFICATION**



**4.1 Workflow**

We started by taking 90% of the data as training, and using 3-fold Cross Validation to find the best hyperparameters. We took a brute-force approach on the selection of algorithm and their hyperparameters. The datasets were not huge and we could test a wide-variety of each, the loop only taking a few minutes. We took the best models for each dataset, then ensembled them to allow voting for the best choice of label.

**4.2 Results**

All models were based on the argmax votes of 5 classifiers(KNN, Linear SVM, SVM with RBF Kernel, Decision Tree, and Random Forest) with tuned hyperparameters listed below. Any unlisted hyperparameters were the default of the sklearn library.

Dataset 1- *80%*

KNN(k = 5), Random Forest(number of estimators = 200)

Dataset 2 - *100%*

KNN(k = 5), Random Forest(number of estimators = 150)

Dataset 3 - *35%*

KNN(k = 14), Random Forest(number of estimators = 175)

Dataset 4 - *88%*

KNN(k = 6), Random Forest(number of estimators = 175)

Dataset 5 - *67%*

KNN(k = 11), Random Forest(number of estimators = 50)

**5. CHALLENGES**

1. Could not get better results from Dataset 3 regardless of effort. Linear regression would get an accuracy of 47% but because it is not a classifier it was not used.
2. Tried more advanced machine learning algorithms such as XGBoost but they did not increase accuracy. Perhaps from a lack of understanding of the algorithms and their hyperparameters. We stuck with the algorithms we knew so we could better trust our results.
3. Better MSE may provide better results. We were able to get 2% better performance on two datasets by switching from linear interpolation to mean imputation.

**6. REFERENCES**

[Scikit-learn: Machine Learning in Python](http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html), Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Jones E, Oliphant E, Peterson P, *et al.* **SciPy: Open Source Scientific Tools for Python**, 2001-, <http://www.scipy.org/> [Online; accessed 2019-04-26].