1.0 Abstract

In this mini project, our group implemented and investigated the performance of a multi-class logistic regression algorithm, specifically softmax regression, in tandem with a mini-batch optimization algorithm: gradient descent with momentum. This was compared to KNN, another regression classifier technique studied in this course. The datasets used were Scikit-Learn Digits Dataset and OpenML's wine-quality-white dataset. Results showed that using 5-fold cross validation and Simple GridSearch CV techniques, softmax regression scores were better for the digits dataset and poore for the wine dataset in comparison to KNN regression accuracy. Different parameters of the optimization function resulted in different performances, as visualized through training and validation curves.

2.0 Introduction

This mini-project report is presented the implementation multi-class logistic regression algorithm. The two data sets selected were Digits (directly available in Scikit-Learn) and White Wine quality (OpenML datasets). This mini-project objective was to develop a Softmax regression model and compare it against one of several other classification algorithms, in our case K-NN. One of the challenges faced in this project was to develop the entire code from scratch and following the equations and mathematical procedures discussed in lectures to gain experience implementing codes. Different hyper-parameters optimizations were included, such as mini-batch gradient descendant, simple grid search and K-Fold.

3.0 Datasets

The datasets used were the Scikit Learn Digit Dataset and the Open ML 'wine-quality-white' dataset. The digits dataset has 10 classes, and 1797 samples, with each feature being an integer between 0 and 16. The wine dataset had 7 classes and 4898 samples, with each of its 11 features corresponding to a different property of the wine (fixed acidity, citric acid, pH, etc.)[1] Normalization on the digits dataset was underdone to help refine the model and optimization: specifically mean-centering and setting the Euclidean norm to 1.0). To complete the 5-fold CV technique, the total rows of each dataset were first shuffled, then split into 5 distinct entities in order for training and testing to be completed correctly.

Certain variables were one-hot encoded to convert the categorical data into integer data. It is important to note that in the OpenML wine dataset, for digits data the descriptive statistics showed some imbalances (zero and near-zero variance); then feature variables with less than 10% variance was removed. In case of wine, the attributes are dependent on each other, and may be correlated, hence in some circumstances it may be applicable to utilize feature selection in regression tasks. Multicollinearity in the wine dataset feature variables (e.g., multiple acidity +/- alcohol variables, same for sulfur (total + dissolved or something). For the multicollinearity, a method such as partial least squares regression could be an alternative, due this maximizes the covariance between latent structures (orthogonality) and considers both the feature and target variable structures by the singular value decomposed of a correlation/covariance matrix.

Both sets were pre-processed using code computed manually, the first step was centered resting the mean to each feature, then rescaling each variable by setting L2-norm to 1. After, we insert leading bias

[1] https://www.openml.org/d/40498

term (w0=1) column to features data frame. Finally, a “One-hot” encode was applied, to convert the target in a multiclass target variable for classification algorithm.

In the next stage a Softmax Multiclass Regression Classifier was coded from scratch. A Softmax regression (or multinomial logistic regression) is a generalization of logistic regression to the case where we want to handle multiple classes. Firstly, a mini-batch stochastic gradient descent was coded, this procedure is mixture Batch Gradient Descent, which converges directly to minima; and a stochastic gradient decedent (SGD) converges faster for larger datasets. Performing this, helps us achieve the advantages of both the former variants. Additionally, a momentum using Nesterov accelerated gradient (i.e., 'look ahead' momentum) with an early stopping mechanism triggered by model loss crossing our epsilon threshold was added in the class definitions.

4.0 Results

4.1 Multiclass (Softmax) Logistic Regression

For the digits dataset, the Softmax Regression Model results has a high accuracy identifying the manually written digits, with a performance close to 90% in each Digit, with an initiation parameters alpha=0.01 (learning rate), epsilon=1e-4 (threshold), epochs=1000 (iterations), batch_size=8, beta_decay=0.9, and early_stop_iter=100 (early stopping). For example, class 2 reported an accuracy of 97% (Figure 1). A simple gridsearch optimization was performed for the alpha hyperparameter (learning rate). This optimization shows that the model fitted to the digits dataset achieves its best performance with a value of 0.01 (Tables 1-3). Figure 3 clearly shows the best results from a 5-fold cross-validation (CV) test for this alpha value, returning the highest accuracy (91%) and the lowest RMSE (1.524).

Relative to the digits dataset, the multiclass (softmax) regression model performed much more poorly for the wine dataset. Maximum mean training accuracy, validation (testing) accuracy and RMSE values of 38.3%, 38.0% and 1.087, respectively, are reported (Tables 4-6) from a model initiated with parameter values of alpha=0.001, epsilon=1e-5, epochs=1000, batch_size=32, beta_decay=0.9, and early_stop_iter=100. These results are also visualized in Figures 4. It was determined that this poor performance is partially due to an imbalanced dataset that would greatly benefit from a stratified K-fold CV approach. This was unofficially implemented making use of the sklearn library (improved training and testing accuracies up to ~55%) however coding difficulties for a ‘from scratch’ implementation impeded this effort (and was therefore left out of the attached code model). Regardless, these accuracies remain relatively low and are likely due to a certain level of multicollinearity between the assumed ‘independent’ feature variables. An improved modelling approach could explore other methods such as Partial Least Squares (PLS) regression, which is known to better handle multicollinearity issues by maximizing covariance between latent structures (orthogonality). PLS regression is similar to Principal Components Analysis, however is considered an improvement by considering the relationship between both independent feature and dependent target variables through the singular value decomposition (SVD) of a correlation/covariance matrix between the two.

4.2 K-Nearest Neighbor (KNN) Model Comparison

When running KNN regression on the datasets RMSE and accuracy were utilized to determine the correct tuning of the hyperparameter: number of nearest neighbors. For each training and validation set, the algorithm ran through a loop of potential of nearest neighbors (set from 1 to 25) to determine RMSE, accuracy, and training and validation scores. It is important to note that 5-fold CV was utilized with a random shuffle before the KNN regression was run. Referring to Figures 5 (A & B) , it is evident that for the Digits dataset, the best tuning of the KNN would be for around 3-4 neighbors, as this yielded the

lowest RMSE and highest accuracy. Referring to Figures 5 (C &D), it is evident that for the wine dataset, the best tuning of the KNN would be around 3-4 neighbors as well (Appendix, Figure 5), as it yielded the lowest RMSE. However, accuracy is quite low for this dataset, evidence that the features may be correlated. This evidence lines up well with the results of the SoftMax regression algorithm with gradient descent optimization, the regression tasks yield similar results.

5.0 Discussion and Conclusions

In this project, we develop a SoftMax regression code with Hyper-Parameter optimization and compare the performance with another algorithm (KNN). In both algorithms, the accuracy and RMSE was very similar. The multi-class regression algorithm did not have good performance in Wine's Quality dataset due to imbalanced data that would have benefitted from stratified k-fold. Perhaps exploring other methods, such as Principal Component Analysis (PCA) or Partial Least Squares (PLS) regression would have further improved modelling performance.

6.0 Statement of Contributions

Dean Meluban
Tasks: Data acquisition, preprocessing, model fitting (KNN: digits/wine data), report writing.

Javier Ordenes Alegria
Tasks: Data acquisition, preprocessing, model fitting (multiclass: digits data), report writing.

Ryan Wilson
Tasks: Data acquisition, preprocessing, model fitting (multiclass: digits data), report writing.

7.0 Appendix

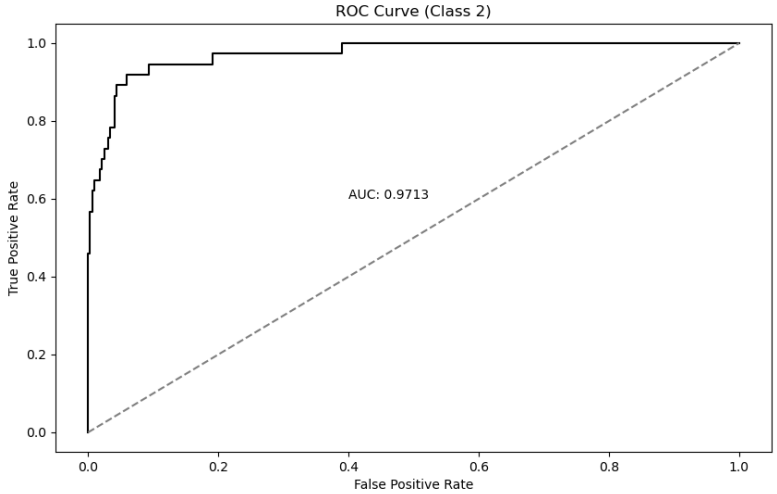

Figure 1: ROC Curve for Digits Data Set (Multiclass Logistic Regression).

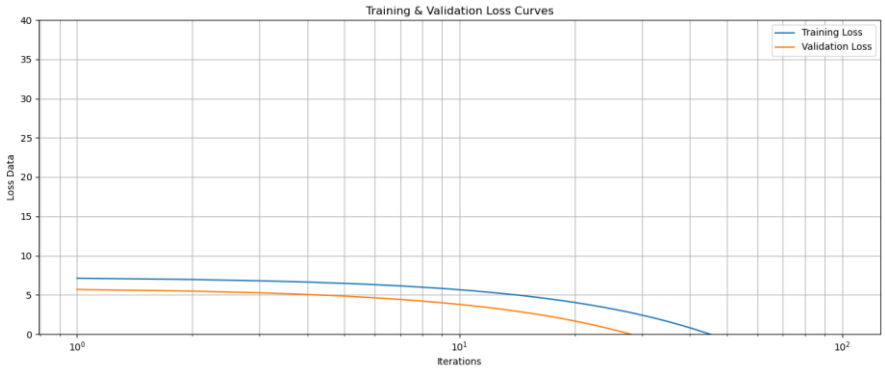

Figure 2: Training and Validation Loss Curves for Digits Data Set (Multiclass Logistic Regression).

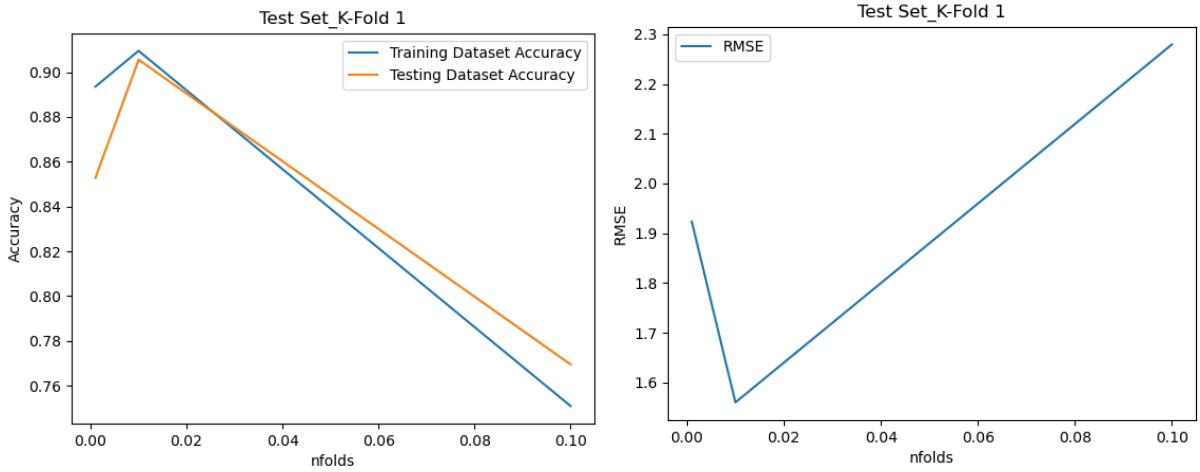

Figure 3: Alpha Hyperparameter Optimization by Simple Grid Search for digits dataset (Multiclass Logistic Regression).

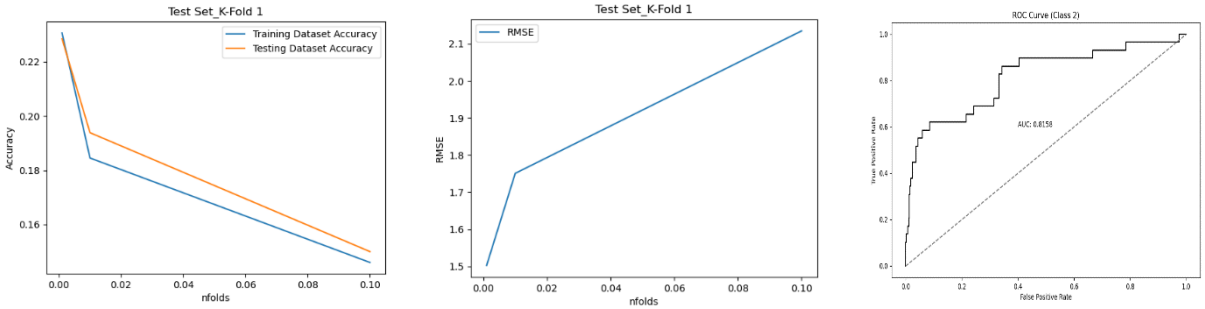

Figure 4: Alpha Hyperparameter Optimization (a, b) by Simple Grid Search & ROC Curve for wine dataset (Multiclass Logistic Regression).

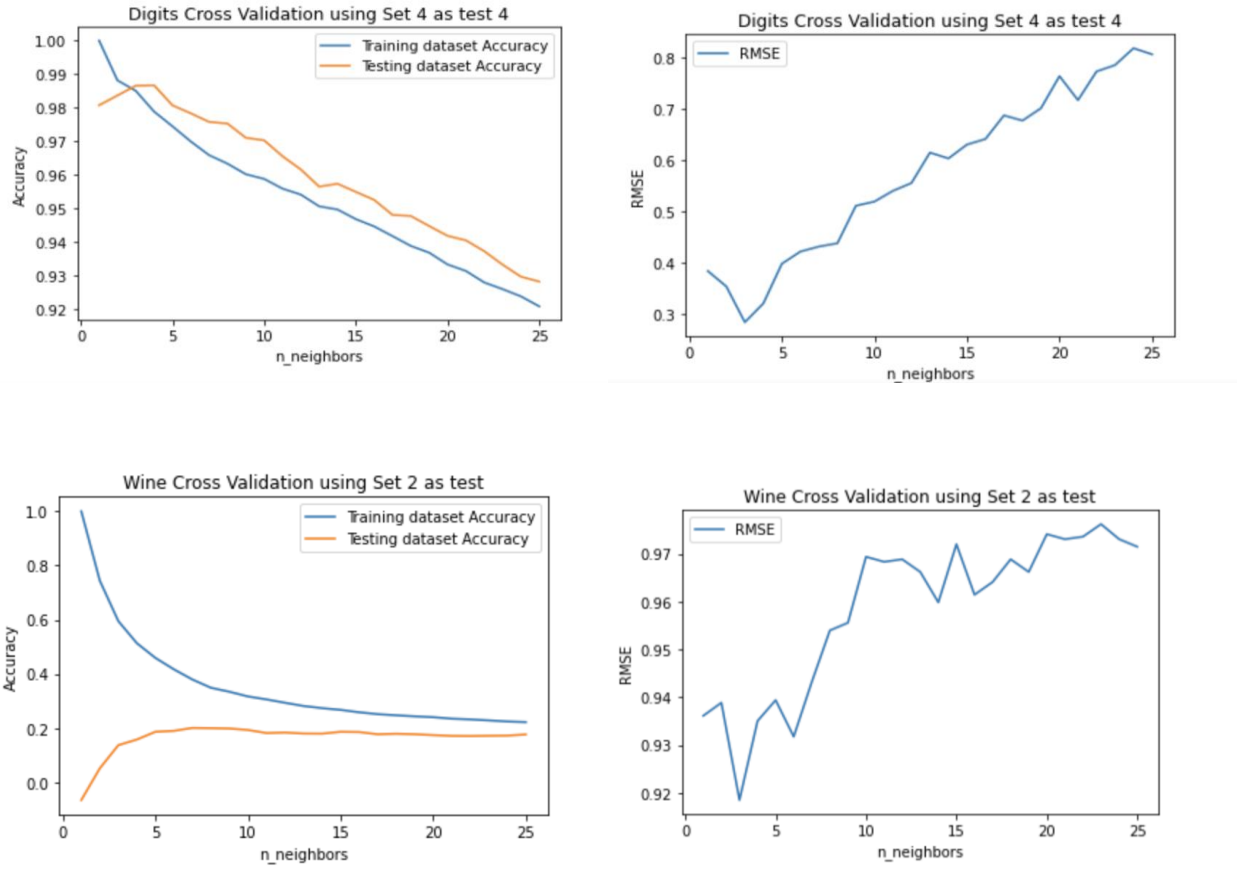

Figure 5: KNN regression Model results. A) Digits Dataset Cross Validation Training- Accuracy Test. B) RMSE Digits Dataset Cross Validation. C) Wine Dataset Cross Validation Training- Accuracy Test. D) RMSE Wine Dataset Cross Validation.

Table 1. Softmax train accuracy for varied alpha values (digits data).

Alpha =	0.1	0.01	0.001	Mean (nFold)
nFold 1	0.778	**0.916**	0.892	**0.862**
nFold 2	0.745	**0.912**	0.893	**0.850**
nFold 3	0.789	**0.918**	0.886	**0.864**
nFold 4	0.765	**0.913**	0.894	**0.857**
nFold 5	0.778	**0.914**	0.887	**0.859**
Mean (Digits)	0.771	**0.915**	0.890	**0.859**

Table 2. Softmax test accuracy for varied alpha values (digits data).

Alpha =	0.1	**0.01**	0.001	Mean (nFold)
nFold 1	0.797	**0.906**	0.883	**0.862**
nFold 2	0.781	**0.906**	0.867	**0.851**
nFold 3	0.778	**0.911**	0.894	**0.861**
nFold 4	0.739	**0.883**	0.861	**0.828**
nFold 5	0.812	**0.905**	0.896	**0.871**
Mean (Digits)	0.781	**0.902**	0.880	**0.855**

Table 3. Softmax RMSE for varied alpha values (digits data).

Alpha =	0.1	0.01	0.001	Mean (nFold)
nFold 1	2.158	**1.445**	1.524	**1.709**
nFold 2	2.186	**1.448**	1.535	**1.723**
nFold 3	2.242	**1.499**	1.686	**1.809**
nFold 4	2.452	**1.557**	1.836	**1.948**
nFold 5	1.961	**1.669**	1.534	**1.722**
Mean (Digits)	2.200	**1.524**	1.623	**1.782**

Table 4. Softmax train accuracy for varied alpha values (wine data).

Alpha =	0.1	0.01	**0.001**	**Mean (nFold)**
nFold 1	0.095	0.285	**0.384**	**0.255**
nFold 2	0.106	0.264	**0.385**	**0.252**
nFold 3	0.117	0.254	**0.377**	**0.250**
nFold 4	0.101	0.265	**0.385**	**0.250**
nFold 5	0.084	0.226	**0.385**	**0.232**
Mean (Digits)	0.101	0.259	**0.383**	**0.248**

Table 5. Softmax test accuracy for varied alpha values (wine data).

Alpha =	0.1	0.01	**0.001**	**Mean (nFold)**
nFold 1	0.108	0.234	**0.381**	**0.241**

nFold 2	0.067	0.332	**0.381**	**0.260**
nFold 3	0.122	0.335	**0.389**	**0.282**
nFold 4	0.085	0.261	**0.357**	**0.234**
nFold 5	0.098	0.239	**0.393**	**0.243**
Mean (Digits)	0.096	0.280	**0.380**	**0.252**

Table 6. Softmax RMSE for varied alpha values (wine data).

Alpha =	0.1	0.01	**0.001**	**Mean (nFold)**
nFold 1	2.341	1.758	**1.103**	**1.734**
nFold 2	2.477	1.505	**1.103**	**1.695**
nFold 3	2.282	1.475	**1.060**	**1.606**
nFold 4	2.375	1.597	**1.067**	**1.680**
nFold 5	2.372	1.839	**1.100**	**1.770**
Mean (Digits)	2.369	1.635	**1.087**	**1.697**