Data 640 Summer 2022

Professor Steve Knode

Tiania Chan

tiania.chan@gmail.com

Assignment 4: Deep Learning

**Introduction**

Introduction:

The objective of the analysis is to use the fashion MNIST dataset to train the deep learning algorithm to be able to identify each item based on a 28 by 28 pixel picture. The libraries TensorFlow and Keras in Python will be to take the fashion MNIST dataset and run through a convolutional neural network. The libraries Pandas, Numpy, and Matplotlib is used to load and visualize the data.

Dataset:

The dataset, fashion MNIST has a training set of 60,000 examples of 28x28 pixel grayscale images and the test set has 10,000 of these images. We have 10 different classes or categeories of clothing that are identifiable in the dataset.

The 10 different classes or categories of clothing are: tshirt or top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. These are labeled from 0 to 9 in the first column of the dataset. The rest of the columns in the dataset are numeric values of each pixel in the images with a total of 784 pixels with values ranging from 0 to 255. The value of 0 through 255 denote the “darkness” of the pixel whether it is more black or whiter in its position of the image.

**Data Preparation**

The fashion MNIST dataset comes from the Zalando’s article images and is a subset of the original MNIST dataset. It uses the same structure of 28 by 28 pixel image for a total of 784 pixels per image. Since this dataset was previously cleaned, this is to talk about the historical cleaning of the dataset which consisted of removing duplicates in the data to avoid bias in the model. Other than that, no further data preparation was needed.

**Predictive Models**

**Results**

As for the results, the workflow is three part. First, the best SVM model was found after the data preparation was done with splitting to a 70/30 split, replacing values in case anything slipped through as well as ensuring the target variable was binary, running a correlation matrix to find that days\_passed was a highly correlated variable and therefore deleted. As shown in Appendix C, Models 1 through 4 were run and the polynomial SVM was chosen. The reason is shown in Appendix E which shows that the Polynomial SVM and Kernal SVM had the best ROC index at .89 training and .91 validation with the smallest variation in the misclassification rate at .11 for both training and validation sets. However, ensemble nodes cannot be run with an active set which is why Polynomial SVM was chosen due to it being an interior point SVM which the ensemble node by SAS miner can and will run. Therefore, shown in Appendix C, the polynomial SVM was run along with models 6, 7, 8, 9, 10, 11, and 12. Since models 6 through 9 were freestanding ensemble models, ensemble models 10, 11, 12 along with model 4 were run through the ensemble nodes which is reflected in Appendix D as Models 15 through 18. From there, the model comparison node was ran and the results are shown in Appendix F where part 1 is a table showing the ROC index and misclassification rate of each of the models and ensemble node models. Appendix F part 2 shows the results of the confusion matrix, the counts for false negative, true negative, false positive, true positives of each of the models as well as the precision rate and ROC index and misclassification rates of each of the models. Appendix F shows that the Random Forest model performed best out of all the models, so a further step was taken to see if skew adjustment was needed for the target variable or not. Appendix F shows the results of the random forest model with a cut off of .5 as well as an optimal cutoff of .15. However, the random forest model performed so well that the cutoff nodes were not needed and there was no effect to the prediction results.

**Conclusion**

For this analysis, different ensemble and classification methods were run on the bank marketing dataset to predict whether new customers would subscribe to the product or not. The different classification models ranged from neural networks, SVM models, to decision trees. The ensemble methods were run using bagging, boosting, random forest, and gradient boosting. The last approach was to use the ensemble node to look at the different variations of posterior probabilities. The conclusion of this study or workflow is that a random forest model with a 70/30 split with the same data preparation as all other models does the best with a ROC index of .93 training, .91 validation, a misclassification rate of .09 training, .1 validation with the precision rate hovering in the low 70s at .75 training and .71 validation. It is the most accurate model while still being sensitive enough to the changes in the dataset.

Next steps in this project would be to try using a dense neural network as well to compare results. A more in depth comparison can be done by using Pandas to create a data frame and run statistical analysis on the results as well for more insight to the results. Also, creating a confusion matrix based on the results of the matrix for more statistical variables to look at each of the models to make a more informed decision as to which model would be better for further input after training the model on the fashion MNIST dataset.

**References**

Knode, S. (2017. February 7). Dataset explore[Video]. Vimeo https://vimeo.com/202932134

Keras. (n.d.). Keras API Reference. Retrieved from <https://keras.io/api/>

Karas. (n.d.) Fashion MNIST dataset, an alternative to MNIST. Retrieved from https://keras.io/api/datasets/fashion\_mnist/

Xiao, H. Rasul, K., Vollgraf, R. (2017, September 15). Fashion-MNIST: a novel Image Dataset for Benchmarking Machine Learning Algorithms. Retrieved from <https://doi.org/10.48550/arXiv.1708.07747>

Zalando Research. (2017, Dec 6). Fashion MNIST: An MNIST-like dataset of 70,000 28x28 labeled fashion images. Kaggle.com Retrieved from https://www.kaggle.com/datasets/zalando-research/fashionmnist/metadata

**Appendix**

**Appendix A: Categorical Variables**

**Table

Description automatically generated**

**Appendix B: Numerical Variables**

**Table

Description automatically generated**

**Appendix C: Ensemble Model Diagram**

**Chart

Description automatically generated**

**Appendix D: Models Used**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model Number | Model Type | Data Partition | Replacement | Skew Adjustment | Selection Criterion |
| 1 | Sigmoid 1 | 70/30 | Yes | No | Validation Misclassification |
| 2 | RBF SVM | 70/30 | Yes | No | Validation Misclassification |
| 3 | Sigmoid -1 | 70/30 | Yes | No | Validation Misclassification |
| 4 | Polynomial SVM | 70/30 | Yes | No | Validation Misclassification |
| 5 | Kernel SVM | 70/30 | Yes | No | Validation Misclassification |
| 6 | Gradient Boosting | 70/30 | Yes | No | Validation Misclassification |
| 7 | Random Forest | 70/30 | Yes | No | Validation Misclassification |
| 8 | Boosting Ensemble | 70/30 | Yes | No | Validation Misclassification |
| 9 | Bagging Ensemble | 70/30 | Yes | No | Validation Misclassification |
| 10 | Decision Tree | 70/30 | Yes | No | Validation Misclassification |
| 11 | Regression | 70/30 | Yes | No | Validation Misclassification |
| 12 | Neural Network | 70/30 | Yes | No | Validation Misclassification |
| 13 | Random Forest Cut off .5 | 70/30 | Yes | Yes | Validation Misclassification |
| 14 | Random Forest Cut off .15 | 70/30 | Yes | Yes | Validation Misclassification |
| 15 | Ensemble Vote prop | 70/30 | Yes | No | Validation Misclassification |
| 16 | Ensemble Vote avg | 70/30 | Yes | No | Validation Misclassification |
| 17 | Ensemble Avg | 70/30 | Yes | No | Validation Misclassification |
| 18 | Ensemble Max | 70/30 | Yes | No | Validation Misclassification |

**Appendix E: SVM Model Results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Kernel | Poly | Sig -1 | Sig 1 | RBF |
| ROC Index | Train | 0.89 | 0.89 | 0.97 | 0.97 | 0.97 |
| Mixclassification Rate | Train | 0.11 | 0.11 | 0.04 | 0.04 | 0.04 |
| ROC Index | Valid | 0.91 | 0.91 | 0.86 | 0.86 | 0.86 |
| Mixclassification Rate | Valid | 0.11 | 0.11 | 0.12 | 0.12 | 0.12 |

**Appendix F: Ensemble Model Results**

Part 1:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Neural | Gradient Boosting | E Vote Average | E Vote Prop | Regression | E Max |
| ROC Index | Train | 0.91 | 0.9 | 0.91 | 0.74 | 0.9 | 0.9 |
| ROC Index | Valid | 0.9 | 0.89 | 0.9 | 0.74 | 0.9 | 0.9 |
| Mixclassification Rate | Train | 0.1 | 0.09 | 0.1 | 0.09 | 0.1 | 0.11 |
| Mixclassification Rate | Valid | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 | 0.09 |
|  |  | Decision Tree | Random Forest | E avg | Poly SVM Bagging | Bagging | Boosting |
| ROC Index | Train | 0.81 | 0.93 | 0.91 | 0.89 | 0.86 | 0.97 |
| ROC Index | Valid | 0.79 | 0.91 | 0.9 | 0.91 | 0.86 | 0.88 |
| Mixclassification Rate | Train | 0.1 | 0.1 | 0.11 | 0.11 | 0.11 | 0.2 |
| Mixclassification Rate | Valid | 0.1 | 0.11 | 0.11 | 0.11 | 0.11 | 0.21 |
|  |  | Random Forest .5 cutoff | Random Forest .15 cutoff |  |  |  |  |
| ROC Index | Train | 0.93 | 0.93 |  |  |  |  |
| ROC Index | Valid | 0.91 | 0.91 |  |  |  |  |
| Mixclassification Rate | Train | 0.1 | 0.1 |  |  |  |  |
| Mixclassification Rate | Valid | 0.11 | 0.11 |  |  |  |  |

**Part 2:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model Description | Data Role | False Negative | True Negative | False Positive | True Positive | Precision | ROC Index | Misclassification Rate |
| Neural | TRAIN | 208 | 2677 | 122 | 157 | 0.562724 | 0.91 | 0.1 |
| Neural | VALIDATE | 86 | 1164 | 37 | 70 | 0.654206 | 0.9 | 0.09 |
| Regression | TRAIN | 240 | 2734 | 65 | 125 | 0.657895 | 0.9 | 0.1 |
| Regression | VALIDATE | 105 | 1178 | 23 | 51 | 0.689189 | 0.9 | 0.09 |
| **Random Forest** | **TRAIN** | **309** | **2780** | **19** | **56** | **0.746667** | **0.93** | **0.1** |
| **Random Forest** | **VALIDATE** | **136** | **1193** | **8** | **20** | **0.714286** | **0.91** | **0.11** |
| **Random Forest .5 cutoff** | **TRAIN** | **309** | **2780** | **19** | **56** | **0.746667** | **0.93** | **0.1** |
| **Random Forest .5 cutoff** | **VALIDATE** | **136** | **1193** | **8** | **20** | **0.714286** | **0.91** | **0.11** |
| **Random Forest .15 cutoff** | **TRAIN** | **309** | **2780** | **19** | **56** | **0.746667** | **0.93** | **0.1** |
| **Random Forest .15 cutoff** | **VALIDATE** | **136** | **1193** | **8** | **20** | **0.714286** | **0.91** | **0.11** |
| Gradient Boostin | TRAIN | 255 | 2756 | 43 | 110 | 0.718954 | 0.9 | 0.09 |
| Gradient Boosting | VALIDATE | 107 | 1182 | 19 | 49 | 0.720588 | 0.89 | 0.09 |
| Poly SVM | TRAIN | 305 | 2767 | 32 | 60 | 0.652174 | 0.89 | 0.11 |
| Poly SVM | VALIDATE | 133 | 1187 | 14 | 23 | 0.621622 | 0.91 | 0.11 |
| Decision Tree | TRAIN | 216 | 2702 | 97 | 149 | 0.605691 | 0.81 | 0.1 |
| Decision Tree | VALIDATE | 97 | 1165 | 36 | 59 | 0.621053 | 0.79 | 0.1 |
| Ensemble Vote Prop | TRAIN | 214 | 2715 | 84 | 151 | 0.642553 | 0.74 | 0.09 |
| Ensemble Vote Prop | VALIDATE | 98 | 1172 | 29 | 58 | 0.666667 | 0.74 | 0.09 |
| Ensemble Vote Avg | TRAIN | 238 | 2734 | 65 | 127 | 0.661458 | 0.91 | 0.1 |
| Ensemble Vote Avg | VALIDATE | 104 | 1179 | 22 | 52 | 0.702703 | 0.9 | 0.09 |
| Ensemble Avg | TRAIN | 305 | 2767 | 32 | 60 | 0.652174 | 0.91 | 0.11 |
| Ensemble Avg | VALIDATE | 133 | 1187 | 14 | 23 | 0.621622 | 0.9 | 0.11 |
| Ensemble Max | TRAIN | 197 | 2661 | 138 | 168 | 0.54902 | 0.9 | 0.11 |
| Ensemble Max | VALIDATE | 82 | 1155 | 46 | 74 | 0.616667 | 0.9 | 0.09 |