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 bank marketing campaign dataset and use the surveyed variables to predict whether a customer will subscribe the advertised product in the next marketing campaign. Multiple ensemble models will be developed and compared to SVM models and other methods of ensemble models to find the model with the best overall average to be used for the next marketing campaign. The target variable is subscribed deposit which has been made a binary target variable with values of yes or no. SAS Enterprise Miner will be used to run and compare multiple ensemble models to see which will fit our data the best and give the best predictions.

Dataset:

The dataset, bank marketing campaign has 4,521 rows of data and 16 input variables.

Categorical variables are as follows: contact\_type which the most popular value was cellular at 64.06%, education which most popular was secondary at 51.01%, has credit in default which was heavily skewed towards no at 98.32%, has housing loan which was relatively even with yes at 56.6%, has personal loan which was majority 84.72%, job with management being the most popular at 21.43% closely followed by blue collar at 20.92%, last contact month which had May at the most done at 30.92%, marital status which had married at 61.87% outcome previous campaign which the most popular result was unknown at 81.95%, and lastly subscribed deposit which was mostly no at 88.46%. This is summarized in Appendix A.

Our numerical variables are as follows: Age which had a mean value of 39.77, no missing values, a minimum of -1, a maximum of 871 and the skewness was 2.72. Next, Average credit balance had a mean of 1422.658, minimum of -3313, maximum of 71188, and a skewness of 6.596. Days passed has a mean of 39.77, minimum of -1, maximum of 871, and skewness of 2.72. Last contact day has a mean of 15.92, no missing values, a minimum of 1, maximum of 3, and a skewness of 0.095. Next is last contact duration in seconds which has a mean of 263.96, no missing values, 4 is the minimum value, 3025 is the maximum value, and skewness is 2.77. Number of contacts is the number of times that a customer has been contacted previously. The mean is 2.79, no missing values, minimum value of 1, maximum value of 50, and skewness of 4.74. Last but not least, previous contacts have a mean of 0.54, no missing values, minimum of 0, maximum of 50 with a skewness of 5.88. This is summarized in Appendix B.

The target or output variable that the model will predict is subscribed deposit. The outcome will help predict the success rate of further datasets to see if a customer will subscribe to the deposit offering. It comes in 3 different outcomes: yes, no, and unknown category. The target variable will be transformed to only yes or no to allow for it to be binary and fit into the ensemble models.

**Data Preparation**

There were no missing variables, so no imputation was needed but did need to replace a few variables to clean some unneeded values in the variables as well as get a binary outcome. All changes done for the different ensemble models can be seen in Appendix C. Those changes are as follows: The first variable, has personal loan had three possible values of no at 2689 counts, yes at 477 counts and unknown at none. Unknown was replaced with no. Next, was the variable job which has the values blue collar at 691 counts, management at 661 counts, technician at 544 counts, admin at 329 counts, services at 306 counts, retired at 163 counts, self-employed at 116 counts, entrepreneur at 111 counts, unemployed at 90 counts, housemaid at 72 counts, student at 53 counts, and two unknowns at 30 and 0 counts respectively. Most of these categories were kept but grouped the two unknown categories into one. The next variable, last contact month may seem unimportant but is generally helpful in establishing rapport with a customer had the values may at 987 counts, July at 501 counts, august at 444 counts, June at 365 counts, November at 265 counts, April at 208 counts, February at 157 counts, January at 99 counts, October at 56 counts, march at 35 counts, September at 35 counts, December at 14 counts and unknown. These were all kept but, in the dataset, unknown was formatted as “\_UNKNOWN\_” so it was reformatted to “unknown”. The same action was taken for marital status where the values were married at 1974 counts, single at 826 counts, divorced at 366 counts and unknown. The next variable, outcome previous campaign, had unknown at 2585 counts, failure at 335 counts, other at 152 counts which we replaced to fall under unknown as well, success at 92 counts, and another unknown which had no counts but was still regrouped to unknown just in case. Finally, to make sure the target variable remained binary, subscribed deposit had 2799 counts of no, 365 counts of yes and an unknown category which was grouped into the no category as well. This makes it so there are only two possible outcomes to make the target variable binary.

After replacing the variable values so the dataset had fewer categories or less unknown as well as making subscribed deposit into a binary variable, the data was also partitioned into a test and validation set. The test set was 70% of the data and the last 30% was the validation set. A second ensemble model was also created by partitioning the data into a 60% test set and 40% validation set. Splitting the data and running the model twice on different portions of the dataset are so to avoid overfitting the data by training on the entire dataset and then validating on the same set to allow for the ensemble model to more accurate. By partitioning the dataset, bias can also be avoided in the dataset so that the models will be able to predict the correct outcomes more accurately on new data if we gained more data following the analysis. Finally, a correlation matrix was run on the replaced dataset and noticed that Days Passed and Last contacted were highly correlated so Days Passed was removed to avoid bias from the data.

**Predictive Models**

Multiple models were developed for this study. First, different variations of SVMs were used with the new data preparation step of removing the highly correlated variable, days\_pased. Removing this variable meant less skew or bias in the ensuing models. Each of the SVM models had a 70/30 split in the data which can be broken down in Appendix D. Model 1 was a Sigmoid 1 value SVM where no skew adjustment was done. Next, model 2 used a radial basis function kernel SVM (RBF SVM). Model 3 uses Sigmoid kernels with a -1 value for the SVM. Model 4 uses the polynomial SVM and then finally, Model 5 uses kernel SVM.

Each of the free standing ensemble models used decision trees for bagging, boosting, random forest and gradient boosting. Bagging, also known as bootstrap aggregating, is where a user will specify the number of different decision tree models that will be developed from a dataset without using all the same data for any of the models. This is done by creating smaller datasets out of the training data by using different samples of the same data and then aggregating it all together. For each variation of this ensemble model, a total of 10 trees created and each tree took 20% of the dataset each iteration and they all used the same seed of 12345.

Next, adaboost, or boosting was used as an ensemble model where the first iteration of the decision tree has equal weight for all the variables and then each time a variable is incorrectly predicted, the weight or importance of the variable is changed so it is more important the next time. This is done until most of the cases are predicted accurately before accuracy falls again. For each of the boosting ensemble models, 20 iterations were done, sampling 10% of the data with the random seed set to the default value also of 12345.

Gradient boosting will resample the data and run decision trees that use a weighted average of the previous dataset. Gradient boosting still uses weighted averages to influence the model, but the accuracy is computed and then residual is used to help adjust the model. The same idea as adaboost is used where this is adjusted until the accuracy of the model drops back down. For each of the gradient boosting models, the models will go through 50 iterations with a seed of 12345. The training portion is set to 60% of the set and the number of maximum branches at the end are 2 since the target variable is binary.

The last type of ensemble model that was used was random forest where each decision tree is used from a bootstrapped sample where it is a smaller subset of the inputs or variables, and the results are combined by majority instead of average. With the random forest ensemble model, a maximum of 100 trees were allowed to be used, the seed was 12345, 60% of the data was used for each of the trees in the random forest ensemble model.

Different classification models were also used in combination with SAS miner’s ensemble node which is reflected in the workflow viewable using Appendix C. Whereas free standing ensemble models, gradient boosting, random forest, boosting, and bagging were models 6 through 9, models 10 through 12 are classification models that were run using the ensemble nodes as well as run independently to compare the results. Model 10 was a decision tree. Model

11 was a Regression Model and Model 12 was a neural network. These were combined with a Polynomial SVM model to compare results of these different models with the results obtained from the ensemble nodes.

In this study, 4 different ensemble nodes were used. In model 15, the ensemble node which looks at the posterior results and votes for the best model using a ratio of the models to the group compared to the total models used. Model 16 looks at the posterior results and averages the results from the majority group. This version of ensemble results is used for class targets only. Next, Model 17 uses an average of the posterior probabilities if the target is a class variable and averages the predicted values if the target is an interval scaled variable. (Knode, n.d.) Model 18 looks at the maximum of the posterior probabilities for a class variable and the maximum predicted value of an interval scaled variable. In this case, the target variable is considered a class variable so these models used the results relevant to a class model.

**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.

**References**

Dataaspirant, (2020, December 17). Seven most popular SVM kernels. Retrieved from <https://dataaspirant.com/svm-kernels/#t-1608054630725>

Knode, S, (n.d.) Week 5 lesson guidance – SAS Enterprise Miner Ensemble knode walkthrough Retrieved from <https://learn.umgc.edu/d2l/le/content/685790/viewContent/26248217/View>

Knode, S. (2016. October 11). Adjusting for skewed target population [Video]. Vimeo <https://vimeo.com/186471846>

Knode, S. (2016, Nov 8). Ensemble Models – Ensemble Node [Video] Retrieved from <https://vimeo.com/190769939?embedded=true&source=video_title&owner=6018107>

Knode, S.( 2016 Nov 11). Ensemble models walkthrough – Heterogeneous Models [Video] Retrieved from <https://vimeo.com/191211534?embedded=true&source=video_title&owner=6018107>

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

Knode, S. (2017. February 7). Log in and Library setup for SAS Miner [Video]. Vimeo

Knode, S. (2021. April 29). Model Assessment [Video]. Vimeo https://vimeo.com/543242649

Knode, S. (2020. September 23). SVM – demo with cutoff [Video]. Vimeo <https://vimeo.com/461049071>

Knode, S. (2016, November 1). Ensemble Models – Bagging Boosting Random Forest [Video]. Vimeo Knode, S. (2020. September 23). SVM – demo with cutoff [Video]. Vimeo <https://vimeo.com/461049071>

SAS Software (2016, May 26). Exploring Input Data and Replacing Missing Values | Getting Started with SAS Enterprise Miner [Video] Youtube. https://www.youtube.com/watch?v=EBFtrPKQ0zg&ab\_channel=SASSoftware

**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 |