Utilizing Machine Learning Techniques to Predict Loan Default

Nicollas Cozzolino, Ignacio Ramos Garca, Tyler Janousek and Sara Guglielmetti, Group15

Abstract—The goal of this case is to create a model to predict the probability an individual will default on their loan given both continuous and categorical variable provided within two datasets. The two datasets provided are a training set and a dataset to apply the model each contain 15,000 records of loans. After testing numerous models on the dataset the Random Forest with feature election was chosen as the final model. It is estimated that the model well reach an accuracy of around 78% on the new dataset.

Index Terms—Machine Learning, Modeling, Payment Default, Loan, Supervised Classification Algorithm

1 Introduction

Lending is one of a few industries that directly and indirectly impact all parts of the economy. This massive industry accounts for trillions in loans from mortgage debt to student loans. The value of a loan is based on the probability of an individual or business paying it back. Hence, lending is essentially a large forecasting and machine learning problem. Through the use of information like FICO scores and income as well as alternative data like an individuals digital footprint, a company can more accurately predict an individuals credit worthiness. [9]

The aim of this report is to examine both categorical and continuous variables to determine the probability an individual will default. The 23 variables used include gender, age, education, history of payments, etc. Using A training set of 15,000 records a supervised classification algorithm is developed. The supervised classification algorithm is then applied to the test set to predict the probability of default for 15,000 loans.

2 Models

2.1 Chi-Square Test

Chi-Square test is a statistical hypothesis test in which the sampling distribution of the test statistic is a chi-squared distribution when the null hypothesis is true. The goal of the chi-squared test is to determine if there is a significant difference between the expected frequencies and the observed frequencies. The chi-squared test is constructed from a sum of squared errors. [6]

2.2 Principal Component Analysis (PCA)

The Principal Component Analysis is a statistical procedure which converts observations of correlated variables into principal components. It is one of the most popular methods for feature selection. However, one disadvantage of using PCA is it treats total data and doesnt include class labels. So, PCA is only effective if original variables are correlated. [1]

2.3 Logistic Regression

Logistic Regression is another statistical method which uses one or more independent variable to determine an outcome. This outcome is measured with a dichotomous variable

which in this dataset is 1 and 0. The ultimate goal of performing logistic regression is to find the best fitting model to depict the relationship between the dichotomous variable and the independent variables [5].

$$p(X) = \frac{e^{\beta 0+\beta 1x}}{1+e^{\beta 0+\beta 1X}} \quad (1)$$

Logistic Function

To estimate all the coefficients $\beta 0$ and $\beta 1$, which are unknown, the principle maximum likelihood is applied (see equation) since it has a better statistical properties than the Least square method used in Linear regression. Maximum likelihood parameter estimation is a technique that can be used when assumptions about the probability distribution of the data can be implemented.

$$l(\beta 0, \beta 1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y'_i=0} (1-p(x_i)) \quad (2)$$

Maximum Likelihood

Normally, datasets have multiple features, variables, or predictors. The equation of this regression can be written as follows:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}} \quad (3)$$

Multivariate Logistic Function

Where p is the number of predictors. As there are 23 predictors the multivariate logistic regression will be implemented.

2.3.1 Normal Logistic Regression

The normal logistic regression as a classification model presents linear parameters. These parameters derive from the sum of the squared residuals minimized. When the model residuals follow a Gaussian distribution, this operation provides also the maximum likelihood estimates of the parameters. The estimation of the maximum likelihood parameter is a technique used to make assumptions about the probability distribution of the data. The likelihood function is derived from the theoretical probability distribution and the observed data, a probability statement which can be created on a specific set of parameter values. If there is the identification of two sets of parameters, the one with a larger likelihood will be considered more relevant. If there are two classes the binomial distribution is the more appropriate probability distribution to adopt. In this distribution, there is the single parameter p which represents the probability of an event or a specific class [2].

$$log\left(\frac{p}{p-1}\right) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p \quad (4)$$

2.3.2 Stepwise Logistic Regression

The stepwise logistic regression classification model is used when the result is a binary or dichotomous variable. Stepwise logistic regression involves the stepwise, also called one-by-one, selection of variables providing two main benefits. First, it is a more effective method to screen a larger number of variables. Second, it fits multiple regression equations simultaneously. Both the normal and stepwise logistic regression are two classification methods search for linear or nonlinear boundaries that optimally separate the data. These boundaries are then used to predict the classification of new samples.[]

- Forward
- Backward
- Bidirectional

Forward Selection (with no predictors) chooses a subset of the predictor variables for the final model. Forward stepwise can be used in context of linear regression whether n is smaller than p or n is greater than p.Forward selection is a very attractive approach, because it's both tractable and it provides a suitable sequence of models. Instead, **backword selection** (with all predictors) begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time. In order to perform backward selection, there needs to be more observations than variables because than the least square regression can be performed when n is greater than p. If p is greater than n, the least squares model can not be applied. Forward and backward stepwise selection is not guaranteed to give the best model containing a specific subset

of the p predictors but, it does avoid over fitting. Performing a full search does not always provide the best solution because of the shift in variance and thus in test error. The best subset could have a better model on the training data, however, this doesn't mean it will have an overall better model in the context of the test data. The third type is the **bidirectional stepwise regression**. It works by comparing the AIC (Akaike Information Criterion) improvements from dropping each candidate variable, and adding each candidate variable between the upper and lower bound regressors sets supplied, from the current model, and by dropping or adding the one variable that leads to the best AIC improvement. In this context, the best **AIC value** is represented by the smallest number [5]. The AIC formula is the following:

$$AIC = 2k - 2ln(L) \tag{5}$$

2.4 K-Nearest Neighbors KNN

KNN uses a different approach from the two previous models. It uses a samples geographic neighborhood to predict the samples classification. As in the regression context, KNN for classification predicts a new sample using the K-closest samples from the training set. This 'closeness' is defined by a distance metric, such as the **Euclidean**[eq.(6)] or **Minkowski** [eq. (7)].

The choice of metric depends on predictor characteristics. For each distance metric, it is strictly relevant to recall the original measurement scales of the predictors. The reason for this operation lies on the affection that predictors scale has on the distance calculations outcome. This implies if predictors are on widely different scales, the distance value between samples will be biased towards predictors with larger scales. To allow each predictor to contribute equally to the distance calculation, it is recommended centering and scaling all predictors prior to performing KNN. As in the regression context, to determine the classification of a new sample, the K-closest training set samples are determined via the distance metric. Class probability estimates for the new sample are calculated as the proportion of training set neighbors in each class. The new samples predicted class is the class with the highest probability estimate; if two or more classes are tied for the highest estimate, then the tie is broken at random or by looking ahead to the K + 1 closest neighbor.[2]

$$\left(\sum_{i=1}^{10}(x_{aj} - x_{bj})^2\right)^{\frac{1}{7}} \tag{6}$$

$$\left(\sum_{i=1}^{10}|x_{aj} - xbj|^q\right)^{\frac{1}{q}} \tag{7}$$

2.4.1 Gower's distance

It is applied for mixed data types, namely, databases with continuous, ordinal or categorical variables at the same time. Gowers General Similarity Coefficient S_{ij} compares two cases i and j and is defined as follows [11]:

$$\frac{\sum_k^n(w_{ijk}S_{ijk})}{\sum_k^n w_{ijk}} \tag{8}$$

where: S_{ijk} denotes the contribution provided by the kth variable, and w_{ijk} is usually 1 or 0 depending if the comparison is valid for the kth variable.

2.5 Decision Trees

The main aim of decision trees is to split data into smaller and more homogeneous groups. In this context homogeneity means the nodes of the partition are purer such as they contain a larger proportion of one class in each node. The concept of purity in the classification environment is represented by the maximization of the accuracy or equivalently the minimization of the misclassification error. However, accuracy as a measure of purity is a little misleading in terms of focus priority. The classification has to have as main focus the portioning of the data in a way which minimizes misclassification and not to place samples in one class. The focus is shift from accuracy to purity due to two alternative measures: the Gini Impurity and the Cross Entropy. For the two-class problem, the Gini index for a given node is defined as:

$$GINI(t) = 1 - \sum_j[(w_]j|t)]^2 \tag{9}$$

When a node t is split into k partitions (children i), the quality of split is computed as:

$$GINI_{split} = 1 - \sum_{i=1}^{k} P(i) \times GINI(i) = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i) \quad (10)$$

where

- ni = number of records at child i
- n = number of records at node t

Entropy for a given node is:

$$Entropy(t) = -\sum_{j} P(w_]j|t)]log[P(w_]j|t)] \quad (11)$$

it measures homogeneity of a node.[2]

2.6 Multilayer Perceptron

The Multilayer Perceptron (MLP) or Artificial Neural Network (ANN) can be viewed as a logistic regression classifier where the input is first transformed using a learned non-linear transformation. This transformation projects the input data into a space where it becomes linearly separable. This intermediate layer is referred to as a hidden layer. A single hidden layer is sufficient to make MLPs a universal approximator. However, there are substantial benefits to using many such hidden layers, for instance the very premise of deep learning. Formally, a one-hidden-layer MLP is a function

$$f : R^D \rightarrow R^L \quad (12)$$

where D the size of input vector x and L is the size of the output vector f(x) , such that, in matrix notation:

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x))) \quad (13)$$

with bias vectors $b^{(1)}$,$b^{(2)}$; weight matrices $W^{(1)}$,$W^{(2)}$,and activation functions G and s.

The vector

$$h(x) = \phi(x) = s(b^{(1)} + W^{(1)}x) \quad (14)$$

constitutes the hidden layer.

$$W^1 \in R^{(DxD_h)} \quad (15)$$

The weight matrix connecting the input vector to the hidden layer. Each column W_i^1 represents the weights from the input units to the i-th hidden unit. Typical choices for $s$include $tahn$,

$$tahn(a) = (e^a - e^{(-a)}/(e^a + e^{(-a)}) \quad (16)$$

or the logistic $sigmoid$ function, with

$$sigmoid(a) = 1/(1 + e^{(-a)}) \quad (17)$$

Both the$tahn$ and the $sigmoid$ are scalar-to-scalar functions, but their natural extension to vectors and tensors consists in applying them element. The output vector is then obtained as:

$$0(x) = G(b^{(2)} + W^{(2)}h(x)) \quad (18)$$

Therefore, class-membership probabilities can be obtained by choosing as the function, in the case of multi-class classification.

To train an MLP, all of the parameters of the model are studied, and the Stochastic Gradient model is applied. Specifically, the Stochastic Gradient Descent with mini-batches. The set of parameters to learn is the set

$$\theta = W^{(2)}, b^{(2)}, W^{(1)}, b^{(1)} \quad (19)$$

Obtaining the gradients

$$\partial l/\partial\theta \quad (20)$$

is achieved through a special case of the chain-rule of derivation, the backpropagation algorithm.

2.7 Random Forest

Random Forest is a flexible machine learning algorithm that produces great result most of the time even without hyper-parameter tuning. Indeed It is a supervised learning algorithm which creates a forest and makes it somehow random. The "forestit builds, is an ensemble of Decision Trees, most of the time trained with the bagging method. The general idea of the bagging method is a combination of learning models increases the overall result. Moreover, one big advantage of random forest is it can be used for both classification and regression problems, which form the majority of current machine learning systems. The random-forest algorithm brings extra randomness into the

model, when it is growing the trees. Instead of searching for the best feature while splitting a node, it searches for the best feature among a random subset of features. This process creates a wide diversity, which generally results in a better performing model. The random forest algorithm is part of the bootstrap method. [10]

2.8 Overfitting and Underfitting

Overfitting refers to a model that models the training data too thoroughly. Overtting occurs when a model learns the detail and noise in the training data to an extent which causes it to negatively impact the performance of the model. This means the noise or random fluctuation in the training data is learned as concepts by the model. However, these concepts do not apply to the new data hence it negatively impacts the models ability to generalize. Overfitting occurs frequently with nonparametric and nonlinear models because these models have more flexibility when learning a target function. As such, many nonparametric machine learning algorithms include parameters or techniques to limit and constrain how much detail the model absorbs.

Underfitting is a model that can neither model the training data nor generalize to the new data. An undert machine learning model is not a suitable model. Undertting is often not discussed as it is easy to detect given a good performance metric. The remedy is to try alternate machine learning algorithms. Nevertheless, underfitting does provide a beneficial contrast to overfitting.

2.8.1 Pruning

Pruning is a technique which reduces the size of decision trees by removing sections of the tree that provide little power to classify instance. Pruning reduces the complexity of the nal classifier, and hence improves predictive accuracy by reducing overfitting. In this report the complexity parameter (CP) was applied to control tree growth. If the cost of adding a variable is higher than the value of the CP, the tree growth ceases. The optimal value of the CP was discovered by using the cross-validation error and the xerror. Additionally, this lead to the discover of the pre-pruning parameters and the optimal trees.

Pre-pruning is also known as early stopping criteria. As the name suggests, the criteria is set as parameter values while building the model. The tree stops growing when it encounters any of the pre-pruning criteria listed below or it detects the pure classes.

- **maxdepth:** Used to set the maximum depth of a tree. Depth is the length of the longest path from a Root node to a Leaf node. Growth of the tree will stop when the depth is equal to the value set for maxdepth.
- **minsplit:** The minimum number of records that must exist in a node for a split to happen or be attempted. For example, if minimum records in a split is 5; then, a node can be further split achieving purity when the number of records in each split node is more than 5.
- **minbucket:** The minimum number of records that can be presented in a Terminal node. For example, if the minimum records in a node is 5, every Terminal/Leaf node should have at least ve records. If the minimum number is set to a small value, like 1, there is a risk of overfitting the model. To remedy the overfitting the parameter should be specified.

2.8.2 Dropout technique

Dropout is technique which addresses two main issues: overfitting and combining exponentially different neural network architectures. The term dropout refers to dropping units, hidden and visible, in a neural network. Dropping units refers to temporarily removing it from the network, along with its incoming and outgoing connections. The units chosen to drop is random.

In the simplest case, each unit is retained with a fixed probability p independent of other units. P can be chosen using a validation set or can be set at 0.5, which is near the optimal

for a wide range networks and tasks. For the input units, however, the optimal probability of retention is closer to 1.[7]

2.9 K-Fold Cross Validation

The samples are randomly partitioned into k sets of roughly equal size. Then a model is developed using all the samples expect the first subset, also called the first fold. The held-out samples are predicted by this model and used to estimate performance measures. The first subset is returned to the training set and the procedure repeats with the second subset, and so on.

The k resampled estimates of the performance measure are summarized (usually using the mean and standard error). Additionally, the k resampled estimates are used to understand the relationship between the tuning parameter(s) and model utility. The choice of k is typically 5 or 10, but there is no formal rule. As k increases, the difference in size between the training set and the resampling subsets decrease. As k decreases, the bias of the technique increases (i.ee, the bias is smaller for k=10 and k=5).

Another important aspect of a resampling technique is the uncertainty. An unbiased method is estimating the correct value (e.g., the true theoretical performance) however, this may lead to uncertainty. The resampling procedure may produce a different value however if performed multiple times it will estimate the true value. K-fold cross-validation has high variance compared to other methods and, for this reason, is not as effective. It should be noted for large training sets the potential issues with variance and bias are negligible.

2.10 Confusion Matrix and Accuracy

After programming and fitting the model, the predictions, and the way used to analyze the performance of each model is using the confusion matrix and accuracy. The confusion matrix is a relation between the correct and wrong predictions. To compute the accuracy the number of true negatives and false positives are taken into account.

3 Data Exploration

3.1 Chi-Square Test

The initial dataset displays the payment variables, PAY_0, PAY_1, etc., as negative numbers. The Chi-Square test can not effectively perform with negative numbers so all negative numbers were transformed to 0. To perform this analysis an α value of 5% was chosen because the expected level of accuracy in the model is not higher than 95%. If the chi-square test is higher than 5% the variable is not statistically significant, however, if the value is lower than 5% it is statistically significant and is considered in the model.

3.2 Correlations Matrix

To generate the correlations for the continuous variables the team performed the FactoMineR. The first correlation is between the age and the bill amount. However, the correlation is low, for the first month 0.032, so the trend is going the opposite direction of the original prediction. This finding fits logically given an older person has an ability to pay back the loan and hence a higher probability of paying back the loan than a younger person. The same effect in higher scale can be observed by comparing age and payment amount.

The second correlation was discovered between payment amount and the bill amount. The test shows a lower correlation in the month the individual has to pay. For instance, in the same month 1 for both payment and bill amount, 10.5 is the lowest correlation among all other months of the payment amount variable. This pattern is repeated in each month; therefore, the team concludes that in a month in which the individuals must pay the loan they are less likely to pay the loan.

The third correlation is between the payment amounts compared to each other. The payment amounts are all around 27 and when the analysis is expanded to the bill amounts this correlation increases to 70̇This discovery proves the individual's behavior is much more random then the bank charges.

3.3 Results

In conclusion, by using EDA it was discovered which variables to include in the new model because they are relevant both for the payment and for the delay of payment: sex and education. In addition, the analysis noted that the variable age is in a delicate situation. It is significant for the payment but it is not significant for the delay of payment. All the other variables are not enough relevant to contribute on a better performing model.

4 Feature selection, transformation, and recording

After doing the EDA, the team could notice that a feature selection (removing some features) can be made, in order to reduce the complexity of the models. Thus, this will be implemented in the section 5 in which the team will test different combinations trying to reach the highest accuracy.

Furthermore there are some methods in which the feature selection is embedded, such as Logistic Regression, in which a greedy search is implemented. This regression method aims at integrating or removing a feature in each step according to the effect in the performance. In addition, the methods in which a tree is involved performs a feature selection, such as Decision trees or Random Forest. These points will be deeper explained in the next section.

Concerning the feature transformation, as mentioned before, different combinations will be made to reach the highest accuracy but without forgetting the complexity of the model. Therefore, in some models the implementation of **PCA** as a transformation or preprocessing of the continuous features will be implemented to test if a higher accuracy can be obtained reducing the dimensionality.

5 Modeling and Experiments

First, before fitting the models, the team split the data into ten folds, in order to perform in each algorithm the technique called k-fold cross-validation.

Furthermore, as the target variable is categorical (0 or 1), after the splitting there must be a balance between the number of each class in both sets, in order to avoid wrong predictions.

5.1 Logistic Regression

To perform model assessment, as mentioned before, the team has implemented cross-validation in order to see the variance of the accuracies using different folds as test/validation sets. The variance obtained with this algorithm is **0**, meaning that this is stable. In addition, the mean of the accuracies is **0.7081824** and the variance is **0.005357218**.

5.2 Stepwize Logistic Regression

As explained in the section **??** there are different approaches to compute stepwise variable selection. For this project the team has implemented the three methods to compare the performance of the model with the selected variables. The result from the three approaches is that the variables are the same. These variables are all apart from *BILL_AMT4*, *BILL_AMT5* and *BILL_AMT1*, whose p-values of the Chi-test are the highest. The mean of the k-fold cross-validation **accuracies** obtained is **71.855%** with a variance of **0.0114558**.

5.3 KNN

To implement this algorithm first, the number of neighbors used needs to be decided. To set this metaparameter, the team used k-fold cross-validation as well.

To choose the upper bound, the team used the empirical rule-of-thumb popularized by Duda et al.[1] which states the choice of the maximum is equal to the square root of the number of instances. Thus, as there are in total 15000 observations, this number is 122. Nevertheless, choosing a small number of neighbors can lead to under fitting the model. Therefore the final choice was 28, an inflexion point from where the plot starts to decrease.

Furthermore, this model only works properly with standardized data and continuous

variables. Nevertheless, the team has implemented a function that uses the Gower's distance, explained in section2.4.1, to use all the variables, continuous and categorical. Finally the mean of the **accuracies** obtained with 28 neighbors is **67.97**% with a variance of **0.000122**.

5.4 Multilayer perceptron

To implement the MLP, the team has used only one hidden layer. After that, the number of hidden units per layer must be establish. To do so, the technique of k-fold cross-validation has been used.

Furthermore another technique widely used is this field to avoid overfitting is **dropouts**. That means that you discard a percentage of the weights used in a certain layer. In this case, as only one layer has been implemented, a threshold was also implemented, increasing the accuracy obtained. As an example, using a MLP with 5 hidden units, using a threshold of 0.01 and after 0.1 the accuracy increased 0.12. Nevertheless, if you use a high number as a threshold this can derive to underfitting. Finally, the number of hidden units chosen is 4 and a threshold of 0.1, obtaining an **accuracy** of **75.1**% with a variance of 0.0001056320537 which is almost negligible.

5.5 Random Forest

After running all the models on the training dataset and checking it's accuracy on the test dataset the team have noticed the Random Forest method had a good potential to be the best predicting model; running it with the default parameters gave us a first accuracy around **77%**.

The first mutable parameter changed was the number of trees done by the model, searching for an optimal number of trees to align, computational cost and accuracy. To do so **Figure 1** was plotted showing the misclassification by the number of trees; as which shows, after **350 trees** the model stays pretty stable so this number of trees is sufficient, using the minimal computational capacity and avoiding overfitting, reaching an accuracy around **78.2%**

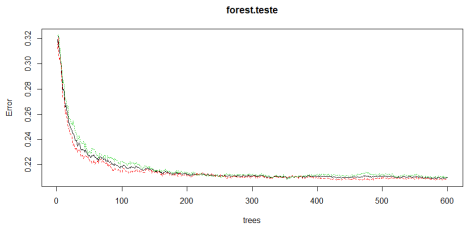

Fig. 1. Mean Misclassification x Number of trees

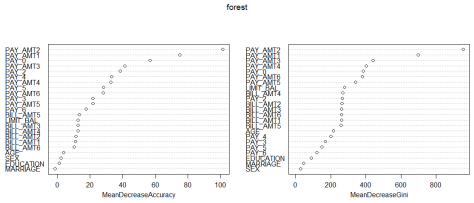

Fig. 2. Mean Decrease of Accuracy and Mean Decrease of Gini-Index per feature

The next step was the **feature selection**, to do so the team has computed the relevance of each variable in terms of decrease in accuracy and Gini-index in case of that feature is taken out as showed in **Figure 2** ; due to this analysis it was discovered the features **EDUCATION** and **MARRIAGE** represented noise in the dataset, and so running the new model without taking into account the noisy features the team have reached the accuracy of **78.6%**.

Keeping searching for improvements in the method the team have plotted a graph showing the OOB (Out of Bagging) error by the Mtry, a parameter which represents the number of variables randomly sampled as candidates at each split, finding the optimal value of **Mtry = 4** as shown in **Figure 3**.

The team has also applied the Random Forest model in a dataset with the categorical features plus 9 of the new features due to the PCA previously ran in the original dataset (9 dimensions), reaching **78.1%** accuracy with 20 features, what was a good result but not better than the one done with feature selection.

6 Conclusion

In conclusion, after test several methods to classify data the **Random Forest** algorithm was chosen to make the predictions on the new data.

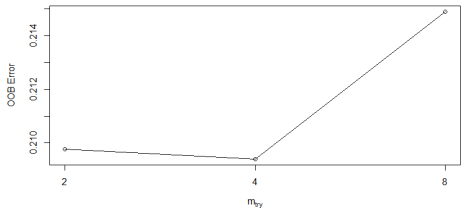

Fig. 3. OOB Error x Mtry

Furthermore, the feature selection version of Random Forest was selected for the final model. The Random Forest was chosen because it had the best accuracy studied, used less features, avoids over-fitting, and is intuitive. Therefore, a tradeoff between complexity and performance is reached. It is estimated the model will reach an accuracy of around 78% on the new dataset.

REFERENCES

[1] Richard O.Duda, Peter E.Hart, David G.Stork *Pattern Classification*, 2nd ed.

[2] Kuhn, M. and Johnson, K. (2016). *Applied Predictive Modeling*, Springer

[3] Hosmer DW, Lemeshow S.*Applied Logistic Regression*,2nd Edition. New York, NY: John Wiley and Sons;2000

[4] Bishop(2006),*Pattern-recognition-and-machine-learning*, Springer-Verlag

[5] J. M. Chambers,T.J. Hastie, *Statistical-Models-in-S*

[6] Alan Julian Izenman *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*, Springer-Verlag New York; 2016

[7] Journal of Machine Learning Research 15 (2014) 1929-1958 *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*

[8] Pre-pruning Classification Trees to Reduce Overfitting in Noisy Domains, Max Bramer 2002

[9] Walker, Jon. Artificial Intelligence Applications for Lending and Loan Management -. TechEmergence, 25 Jan. 2018

[10] *Machine Learning With Random Forests And Decision Trees: A Mostly Intuitive Guide*, Scott Hartshorn, 2016

[11] J. C. Gower.*A General Coefficient of Similarity and Some of Its Properties*,Biometrics, Vol. 27, No. 4. (Dec., 1971), pp. 857-871