Paper SAS4076-2020

Apply the Key Ideas from Andrew Ng's Machine Learning Certification (Coursera) in SAS® Viya®

Petri Roine, SAS Institute Inc.

ABSTRACT

Machine Learning Certification by Stanford University (Coursera) is the single highest rated course on machine learning on the entire internet. Created by artificial intelligence (AI) guru Andrew Ng, co-founder of Coursera and Professor at Stanford University, the program has been taken by more than two million persons globally, who have given it an average rating of a whopping 4.9 out of 5. The course covers how to apply the most advanced machine learning algorithms to common problems. This breakout session takes all the key ideas from Andrew Ng and explains step-by-step how to use those valuable nuggets in SAS® Visual Data Mining and Machine Learning. You will become better at debugging and figuring out how to improve algorithm's performance. We go through supervised and unsupervised learning models and pinpoint what to consider when trying to optimize the performance of the model. You do not have to have taken the actual course—the key ideas are explained at an appropriate level.

INTRODUCTION

Machine learning is the science of getting computers to perform specific tasks without being explicitly programmed. We all use machine learning many times a day, often without even knowing it. Navigators and self-driving cars, speech recognition and smart assistants, web search and image recognition are all everyday examples of how machine learning is now an integral part of our lives (Coursera 2020).

Different industries are using machine learning at a rapid rate and thus the need for domain experts is at an all-time high. This global trend has created an immense need for machine learning training courses and many educational organizations have published their own training catalogs for this domain area.

Machine Learning Certification by Stanford University (Coursera) is the single highest-rated course on machine learning on the entire internet. Created by artificial intelligence (AI) guru Andrew Ng, co-founder of Coursera and Professor at Stanford University, the program has been taken by nearly three million people who have given it an average rating of 4.9 stars out of 5.

A big part of the course is spent on how to implement and deploy machine learning algorithms—something that the SAS® Analytics Platform offers out-of-the-box. The other main theme Andrew Ng used throughout the course was advice on building a machine learning system and deciding what to work on next when the first iteration of the machine learning model is ready.

Interestingly, not everything that was taught in the course would transfer to the SAS world squarely. For example, in the course a lot of focus is on taking small steps in modeling (that is, limiting data sample size and optimizing the learning rate to minimize the number of iterations). On the other hand, the SAS Analytics Platform offers scalable, high-performance in-memory computing built for big data and a capability to train multiple algorithms in a flick of the wrist.

This breakout session presents some of Andrew Ng's best practices from the course and explains how to use those valuable nuggets in SAS® Visual Data Mining and Machine Learning. You will learn how to approach your modeling assignments and how to get up to speed quicker than ever before.

START WITH SIMPLE ALGORITHMS

Andrew Ng suggested to always start modeling with simple algorithms such as linear and logistic regression, and very quickly create a first model without too much feature engineering. This enables you to see very quickly whether the behavior that you want to predict can be modeled and to also get ideas on how to move forward with modeling.

Performing linear regression with a complex set of data with many features is unwieldy. Suppose you wanted to create a model from 100 features and include all the quadratic terms. That would give us 5,050 new features. Andrew Ng advised that neural networks offer an alternative way to perform machine learning when we have complex hypotheses with many features.

Andrew Ng also offered some guidelines for selecting a neural network architecture. He recommended starting with one hidden layer, then three hidden, and then to see which one has the lower cross validation error and choose that architecture. For the number of neurons, he gave a broader criterion: 1 to 4 times the number of features (that is, inputs).

HOW SAS® VIYA® HELPS YOU START WITH SIMPLE ALGORITHMS

The SAS Analytics Platform offers a unique approach to starting simple. SAS® Visual Analytics applications enable you to rapidly create a simple model, apply partitions, and duplicate the model as another algorithm. For example, you can start your classification modeling with logistic regression and duplicate the model as decision tree and neural networks.

Then you can quickly autotune the model's hyperparameters and finally compare models in a Model Comparison object. All of this can be done very quickly to get a good view of the features' predictive power regarding the behavior that we want to model.

Next, the SAS Analytics Platform enables you to create a modeling pipeline with one click. Model Studio opens with your simple model pipeline. SAS offers out-of-the-box best practice model pipelines for three complexity levels: basic, intermediate, and advanced. For example, you can create another pipeline with an intermediate template to get a better understanding of how your data behaves.

All of this can be done in minutes rather than hours or days and gives you a nice advantage over manually creating the pipeline by coding.

UNDERSTANDING BIAS AND VARIANCE

One of the things Andrew Ng taught was that if you run a machine learning algorithm and it does not work as well as you were hoping, almost all of the time it is because you have either a high bias problem or a high variance problem. It was stressed that it is crucial to know whether you have bias or variance or a bit of both. Knowing which of these things is the problem is useful for improving your algorithm.

Bias is the difference between the average prediction of the model and the actual value that we are trying to predict. A model with high bias pays little attention to the training data and oversimplifies the model. It always leads to high errors on training, validation, and test data (Towards Data Science 2020).

Variance is the variability of model prediction for a given data point or a value that tells us the spread of our data. A model with high variance pays a lot of attention to training data

and does not generalize on the data that it has not seen before. As a result, such models perform very well on training data but have high error rates on test data (Towards Data Science 2020).

Underfitting happens when a model is unable to capture the underlying pattern of the data. These models usually have high bias and low variance. Overfitting happens when a model captures the noise along with the underlying pattern in data. These models have low bias and high variance (Towards Data Science 2020).

HOW SAS® VIYA® HELPS YOU UNDERSTAND MODEL BIAS AND VARIANCE

Bias is the accuracy of predictions. A model with high bias has high errors on training, validation, and test data. Model Studio provides many assessment charts and performance metrics out-of-the-box to help evaluate model bias. Figure 1 shows an example of an assessment plot. The Event Classification chart is a visual representation of the confusion matrix at various cutoff values for each partition. The confusion matrix contains four cells that display the counts for true positives for events that are correctly classified (TP), false positives for non-events that are classified as events (FP), false negatives for events that are classified as non-events (FN), and true negatives for non-events that are classified as non-events (TN). True negatives include non-event classifications that specify a different non-event level. From the Event Classification chart, it is easy to see that for our example data and model we get high numbers of incorrect (orange color) predictions in all the partitions that support the assumption of a high bias model.

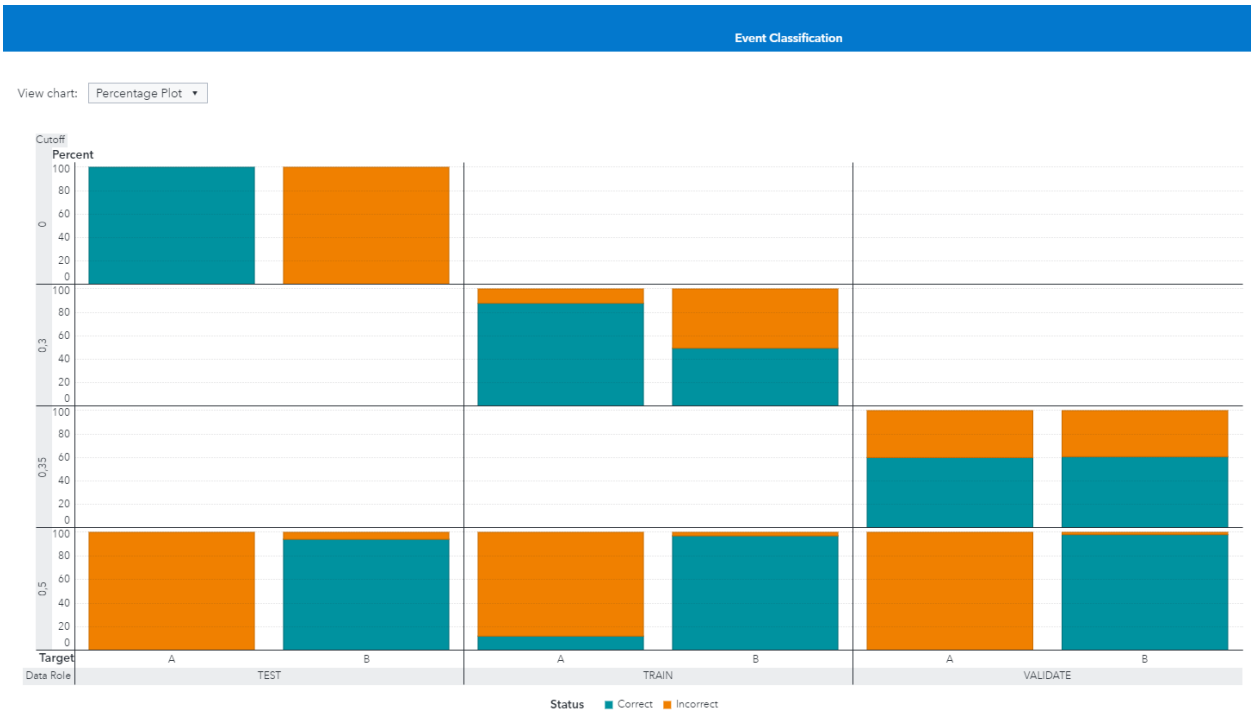


Figure 1. Event Classification Chart

Variance is the amount that the estimate of the target function changes if different training data is used. A model with high variance performs very well on training data but has high error rates on test data. The Model Studio assessment charts and performance metrics also help evaluate model variance. Figure 2 shows an example of an assessment plot. The receiver operating characteristic (ROC) curve is a plot of sensitivity (the true positive rate) against 1-specificity (the false positive rate). Both measures of classification are based on the confusion matrix. These measures are calculated at various cutoff values. From the ROC chart, it is easy to see that for our example data and model, we see a big difference in the

performance with TRAIN data and VALIDATE and TEST data, which supports the assumption of a high bias model.

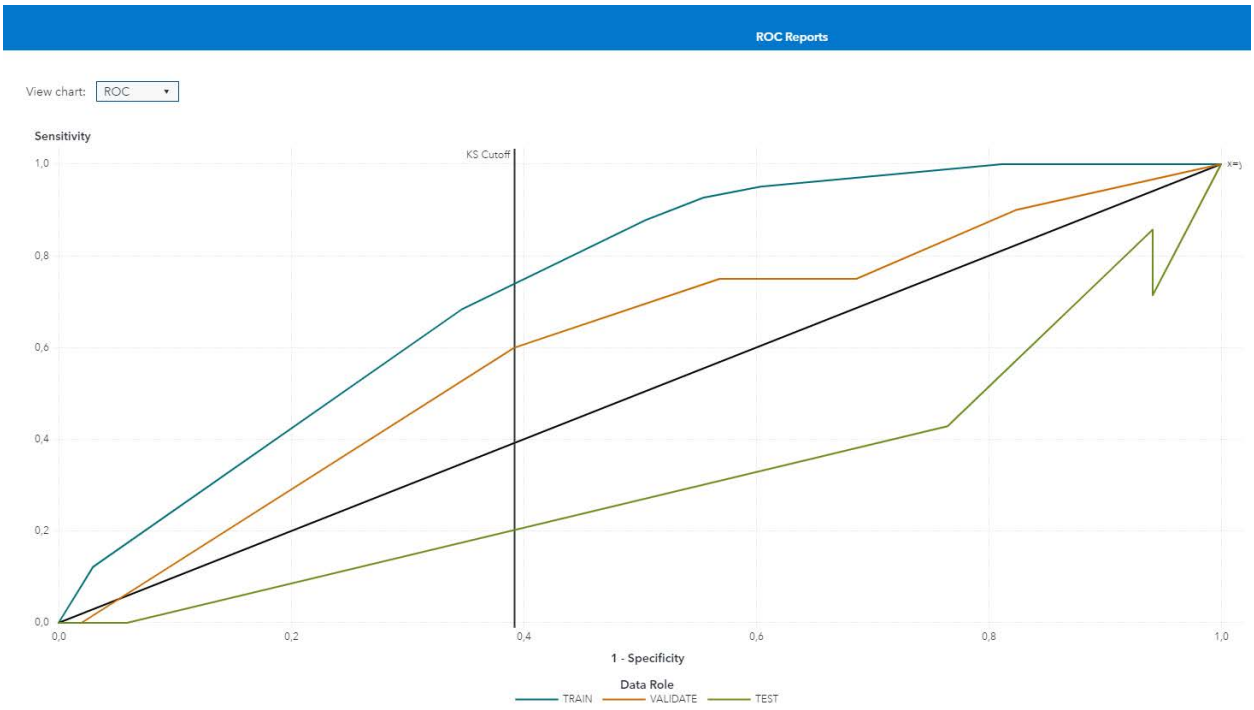


Figure 2. ROC Chart

REDUCE THE NUMBER OF FEATURES

Andrew Ng presented two worthy options to address the issue of overfitting: reducing the number of features and regularization.

Dimension reduction decreases the number of features under consideration. In many applications, the raw data has very high dimensional features, and some features are redundant or irrelevant to the task. Reducing the dimensionality helps find the true, latent relationship. Features can be removed manually, but this is often very time consuming.

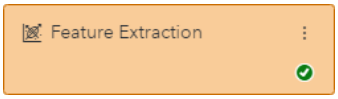
HOW SAS® VIYA® HELPS IN REDUCING THE NUMBER OF FEATURES

Dimension reduction (that is, decreasing the number of variables under consideration) is one of the major functionalities that SAS offers. Model Studio provides you with three nodes in SAS Visual Data Mining and Machine Learning for dimension reduction: Feature Extraction, Variable Clustering, and Variable Selection.

Feature Extraction

The Feature Extraction node creates features (inputs) that encapsulate the central properties of the input data in a low dimensional space. Feature extraction in Model Studio is accomplished using various techniques, including principal component analysis (PCA), robust PCA (RPCA), singular value decomposition (SVD), and autoencoders. You can either select the method, or you can allow the Feature Extraction node to automatically choose the method. Method selection is based on the number of interval input variables. For example, the automatic selection uses PCA when the number of interval inputs is less than or equal to 500.

One drawback to feature extraction is that the composite variables are no longer meaningful with respect to the original problem. It becomes hard to understand what caused the model to come up with the predictions that it created.

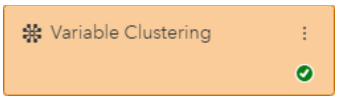


Variable Clustering

The Variable Clustering node divides numeric variables into disjoint clusters and chooses a variable that represents each cluster. In general, the clustering process starts with one variable per cluster, and the number of variables per cluster increases with additional clustering steps. Correspondingly, the total number of clusters decreases with additional clustering steps, because clusters in previous steps are merged together.

In addition, class variables are included in the clustering process, but they are treated differently. The original class variables are replaced by individual binary class level variables (indicator variables with a value of **0** or **1**), with one variable per class level. These class level indicator variables are then included in the clustering process.

The selected representative variable is the variable that contributes the most to the variation in the cluster. Variable clustering removes collinearity, decreases redundancy, and helps reveal the underlying structure of the data set.



Variable Selection

The Variable Selection node uses several unsupervised and supervised methods to determine which variables have the most impact on the model.

Unsupervised selection identifies the set of input variables that jointly explain the maximum amount of data variance. The target variable is not considered with this method. The supervised selection methods that are available are Fast Supervised Selection, Linear Regression Selection, Decision Tree Selection, Forest Selection, and Gradient Boosting Selection.

This tool enables you to specify more than one selection technique, and there are several options for selection criteria. Because there might be disagreements on selected variables when different techniques are used, specifying multiple selection methods helps ensure that important variables are consistently selected. Variables that fail to meet the selection criteria are marked as rejected and not used in successor modeling nodes.

EXAMPLE OF USING DIMENSION REDUCTION NODES

I created a simple test (Figure 3) for comparing all the tree dimension reduction nodes for my data. My data set was Stocks IPO information & results, which is available from Kaggle under CC0: Public Domain license. It is 1,664 columns wide and has 3,762 observations. Data can be downloaded from https://www.kaggle.com/proselotis/financial-ipo-data.

I used a well-known decision tree to keep things simple. In my experience, a decision tree algorithm is highly perceptive to the training data that might cause overfitting. I found this approach interesting because dimension reduction decreases the number of variables under

consideration. I kept all the other default settings for Decision Tree nodes but adjusted the minimum leaf size to 10 instead of 5. I did not use autotuning so that all the decision trees would be run with the same parameters.

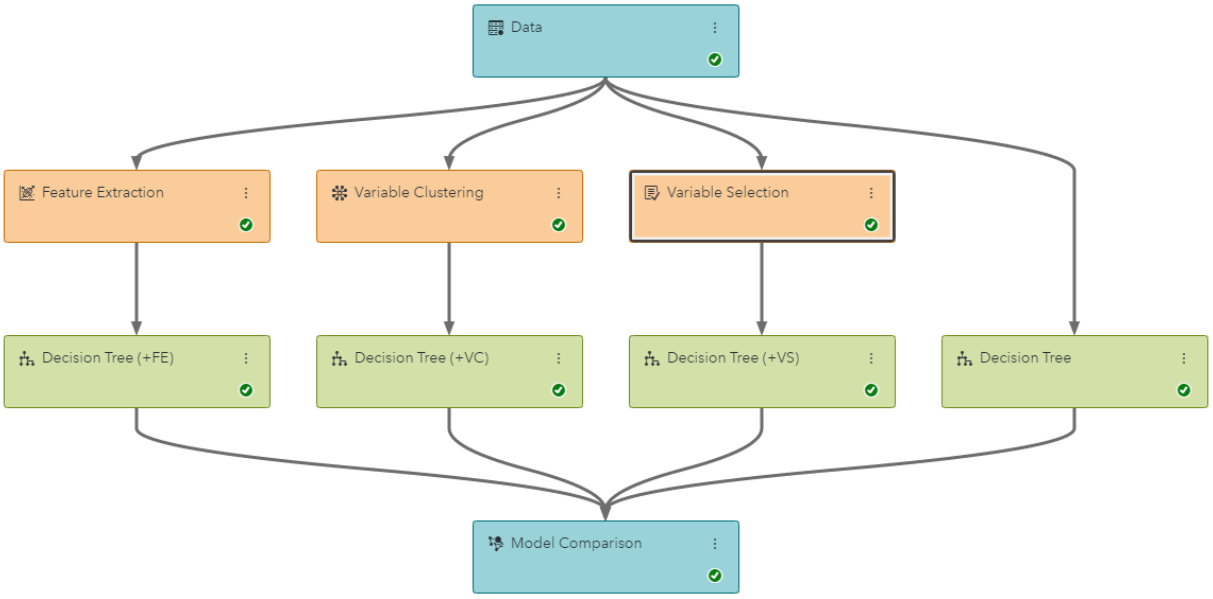


Figure 3. Pipeline to Study Dimension Reduction Nodes

I had a hard time making the Feature Extraction node work with my data. I used several extraction methods and different parameters but did not get good performance. For example, when using the robust principal component analysis (RPCA), it created only two principal components that had a little predictive power, but it seemed to not be able to capture the behavior that was hidden in the data. All of the other created principal components had no significance.

On the other hand, the Variable Clustering node was able to capture a very sensible cluster network with five clusters (Figure 4), but the dimension reduction was small. Out of 1,664 variables, it managed to reduce 55 variables.

The Variable Selection node did a very good job of reducing dimensionality. Out of 1,664 variables, it managed to reduce 1,340 variables.

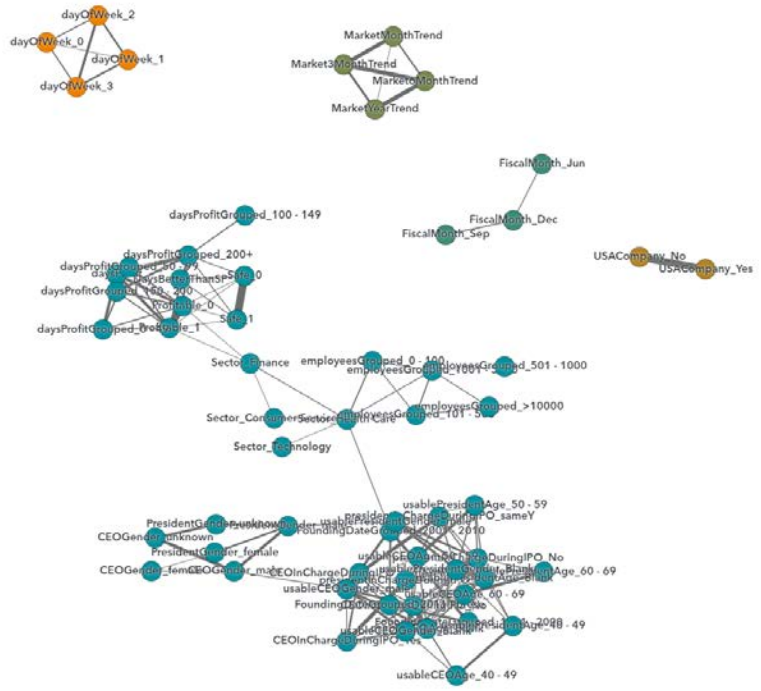


Figure 4. Variable Clustering and Five Clusters

REGULARIZATION

Andrew Ng presented two options to address the issue of overfitting. In the previous section, we discussed dimension reduction. The second suggested option to minimize overfitting is regularization.

Overfitting happens when your model is trying too hard to capture the data points that do not represent the true properties of the data, but rather a random chance. Regularization decreases this complexity by reducing the magnitude of model parameters by penalizing them.

Andrew Ng presented two regularization techniques, Lasso and Ridge, and stated that they work well when you have a lot of slightly useful features. The importance of using regularization can be captured in the famous quote by Owen Zhang, the chief product officer at DataRobot, who said, "If you are using regression without regularization, you have to be very special."

HOW SAS® VIYA® HELPS PREVENT MODEL OVERFITTING WITH REGULARIZATION

Regularization is one of the many functionalities that SAS has made easier for users to apply. Model Studio enables you to manually apply L1 regularization and L2 regularization techniques. However, finding the optimal values for L1 and L2 regularization is often based on trial and error. With the SAS Analytics Platform, you can set the Perform Autotuning property (Figure 5) to on to automatically find the best values for L1 and L2 regularization. Autotuning also adjusts other hyperparameters to the best values. SAS hyperparameter tuning minimizes or maximizes the chosen objective function (typically, a measure of model error). SAS does this by using a preferred method that includes genetic algorithms, grid search, random sampling, Latin hypercube sampling, or Bayesian kriging.

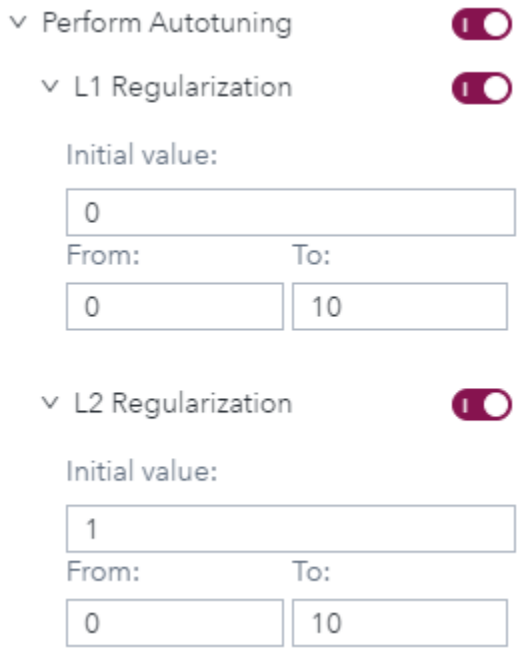


Figure 5. Perform Autotuning to Find Optimal L1 and L2

DECIDING WHAT TO WORK ON NEXT

Andrew Ng also provided good tips for improving your initial model. An example is finding that your model makes large errors when using new data.

1. Get more training examples. This helps if your model has high variance.
2. Try a smaller set of features. This helps if your model has high variance.
3. Try getting additional features or adding polynomial features. This helps if your model has high bias and low variance.
4. Try adjusting regularization parameters. This helps if your model has low bias and high variance (Coursera 2020).

HOW SAS® VIYA® HELPS IMPROVE YOUR MODEL

Getting more training examples and adding additional features is mostly an out-of-the-system activity on the SAS Analytics Platform. The platform is capable of handling very large amounts of wide data, so the model building often starts with all the data rather than a subset of data with selected variables.

Trying a smaller set of features can be easily achieved by either manually rejecting input variables from the user interface or by using dimension reduction nodes that are available in "Overview of Data Mining Preprocessing" in Model Studio.

Adding polynomial features is possible in SAS, but it is also time consuming to produce them. Especially if you have wide data, creating polynomial features becomes burdensome. A better approach could be to use a different algorithm (for example, a neural network or a support vector machine).

Adjusting regularization parameters can also be laborious if done manually. The SAS Analytics Platform comes with an autotuning functionality, which automates the selection of regularization parameters as well as other hyperparameter values using an intelligent optimization-based methodology. This capability can significantly improve your productivity and the accuracy of the resulting model with no additional effort.

CONCLUSION

The Machine Learning (ML) course by Coursera teaches the basics of how to approach machine learning projects. It advises to start with simple models and apply dimension reduction and regularization when needed. It also stresses understanding the model bias and variance to enable you to assess whether more variables and more data are helpful.

The SAS Analytics Platform more than supports all of this, although the approach is different. SAS is built with productivity in mind and many features enable you to almost forget the basics of modeling because they happen behind the scenes. Automated feature engineering selects the best set of features for modeling by ranking them to indicate their importance in transforming your data. Visual best practice pipelines are dynamically generated from your data, yet they are editable to remain as a white box model. All of these automated feature engineering and modeling capabilities enable you to concentrate on the model performance instead of the syntax of your coding language.

REFERENCES

Coursera. "Machine Learning." Accessed January 28, 2020. https://www.coursera.org/learn/machine-learning#about.

Towards Data Science. "Understanding the Bias-Variance Tradeoff." Seema Singh. Accessed February 12, 2020. https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Petri Roine
SAS Institute Inc.
petri.roine@sas.com
https://www.linkedin.com/in/petriroi/