Module 2 – Regression and Prediction

CASE STUDY ACTIVITY TUTORIAL

Case Study 4: Predicting Wages 2

PROFESSIONAL EDUCATION
Digital Programs

Regression 3.3: Assessment of Prediction Quality. Aggregation of Predictors. Case Study

V. Chernozhukov
MIT

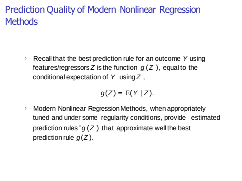
July 25, 2016

Prediction Quality of Modern Nonlinear Regression Methods

- Recall that the best prediction rule for an outcome Y using features/regressors Z is the function $g(Z)$, equal to the conditional expectation of Y using Z,

$$g(Z) = \mathrm{E}(Y \mid Z).$$

- Modern Nonlinear Regression Methods, when appropriately tuned and under some regularity conditions, provide estimated prediction rules $\hat{g}(Z)$ that approximate well the best prediction rule $g(Z)$.



Prediction Quality of Modern Nonlinear Regression Methods

- Recall that the best prediction rule for an outcome Y using features/regressors Z is the function $g(Z)$, equal to the conditional expectation of Y using Z,

$$g(Z) = \mathrm{E}(Y \mid Z).$$

- Modern Nonlinear Regression Methods, when appropriately tuned and under some regularity conditions, provide estimated prediction rules $\hat{g}(Z)$ that approximate well the best prediction rule $g(Z)$.

In this segment, we discuss the quality of prediction that the modern methods provide.

We begin by recalling that the best prediction rule for outcome Y using features/regressors Z is the function $g(Z)$, equal to the conditional expectation of Y using Z.

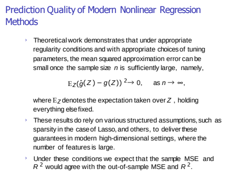
Modern Regression Methods, namely Lasso, Random Forest, Boosted Trees, Neural Networks, when appropriately tuned and under some regularity conditions, provide estimated prediction rules $\hat{g}(Z)$ that approximate the best prediction rule $g(Z)$.

Prediction Quality of Modern Nonlinear Regression Methods

- Theoretical work demonstrates that under appropriate regularity conditions and with appropriate choices of tuning parameters, the mean squared approximation error can be small once the sample size n is sufficiently large, namely,

  $$\mathrm{E}_Z(\hat{g}(Z) - g(Z))^2 \to 0, \qquad \text{as } n \to \infty,$$

  where E_Z denotes the expectation taken over Z, holding everything else fixed.
- These results do rely on various structured assumptions, such as sparsity in the case of Lasso, and others, to deliver these guarantees in modern high-dimensional settings, where the number of features is large.
- Under these conditions we expect that the sample MSE and R^2 would agree with the out-of-sample MSE and R^2.



Prediction Quality of Modern Nonlinear Regression Methods

- Theoretical work demonstrates that under appropriate regularity conditions and with appropriate choices of tuning parameters, the mean squared approximation error can be small once the sample size n is sufficiently large, namely,

$$\mathrm{E}_Z(\hat{g}(Z) - g(Z))^2 \to 0, \quad \text{as } n \to \infty,$$

  where E_Z denotes the expectation taken over Z, holding everything else fixed.
- These results do rely on various structured assumptions, such as sparsity in the case of Lasso, and others, to deliver these guarantees in modern high-dimensional settings, where the number of features is large.
- Under these conditions we expect that the sample MSE and R^2 would agree with the out-of-sample MSE and R^2.

Theoretical work demonstrates that under appropriate regularity conditions... and with appropriate choices of tuning parameters, the mean squared approximation error can be small once the sample size n is sufficiently large: [formula]

These results do rely on various structured assumptions, such as sparsity in the case of Lasso, and others, to deliver these guarantees in modern high-dimensional settings, where the number of features is large.

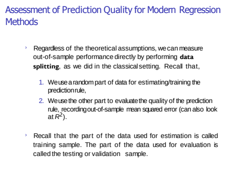
From a practical stand-point, we expect that under these conditions the sample MSE and R^2 will tend to agree with the out-of-sample MSE and R^2.

Assessment of Prediction Quality for Modern Regression Methods

- Regardless of the theoretical assumptions, we can measure out-of-sample performance directly by performing **data splitting**, as we did in the classical setting. Recall that,

  1. We use a random part of data for estimating/training the prediction rule,
  2. We use the other part to evaluate the quality of the prediction rule, recording out-of-sample mean squared error (can also look at R^2).

- Recall that the part of the data used for estimation is called training sample. The part of the data used for evaluation is called the testing or validation sample.



Assessment of Prediction Quality for Modern Regression Methods

- Regardless of the theoretical assumptions, we can measure out-of-sample performance directly by performing **data splitting**, as we did in the classical setting. Recall that,
  1. We use a random part of data for estimating/training the prediction rule,
  2. We use the other part to evaluate the quality of the prediction rule, recording out-of-sample mean squared error (can also look at R^2).
- Recall that the part of the data used for estimation is called training sample. The part of the data used for evaluation is called the testing or validation sample.

We can measure out-of-sample performance directly by performing **data splitting**, as we did in the classical linear regression.
Recall, that we use a random part of data, say half of data, for estimating or training the prediction rules.
Second, we use the other part of data, to evaluate the predictive performance of the rule, recording the out-of-sample MSE or R^2.
Accordingly, we call the first part of data ... the training sample... and the second part ... the testing or validation sample.

- We have a data sample containing observations on outcomes Y_i and features Z_i. Suppose we use n observations for training and m for testing/validation. We use the training sample to compute prediction rule $\hat{g}(Z)$.

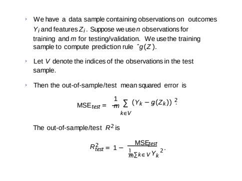
- Let V denote the indices of the observations in the test sample.

- Then the out-of-sample/test mean squared error is

$$\text{MSE}_{test} = \frac{1}{m}\sum_{k\in V}(Y_k - \hat{g}(Z_k))^2.$$

The out-of-sample/test R^2 is

$$R^2_{test} = 1 - \frac{\text{MSE}_{test}}{\frac{1}{m}\sum_{k\in V} Y_k^2}.$$



- We have a data sample containing observations on outcomes Y_i and features Z_i. Suppose we use n observations for training and m for testing/validation. We use the training sample to compute prediction rule $\hat{g}(Z)$.
- Let V denote the indices of the observations in the test sample.
- Then the out-of-sample/test mean squared error is

$$\mathrm{MSE}_{test} = \frac{1}{m} \sum_{k \in V} (Y_k - g(Z_k))^2.$$

The out-of-sample/test R^2 is

$$R^2_{test} = 1 - \frac{\mathrm{MSE}_{test}}{\frac{1}{m} \sum k \in V \, Y_k^2}.$$

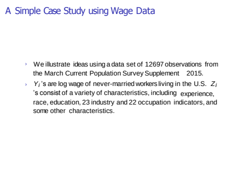
Indeed, suppose we use n observations for training and m for testing or validation. Let capital V denote the indices of the observations in the test sample.

Then the out-of-sample or test Mean Squared Error is defined as the average squared prediction error where we predict Y_k in the test sample by $\hat{g}(Z_k)$, where the prediction rule $\hat{g}$ was computed on the training sample.

The out of sample R^2 is defined accordingly as 1 minus the ratio of the test MSE to the variation of the outcome in the test sample.

A Simple Case Study using Wage Data

- We illustrate ideas using a data set of 12697 observations from the March Current Population Survey Supplement 2015.
- Y_i's are log wage of never-married workers living in the U.S. Z_i's consist of a variety of characteristics, including experience, race, education, 23 industry and 22 occupation indicators, and some other characteristics.



A Simple Case Study using Wage Data

- We illustrate ideas using a data set of 12697 observations from the March Current Population Survey Supplement 2015.
- Y_i's are log wage of never-married workers living in the U.S. Z_i's consist of a variety of characteristics, including experience, race, education, 23 industry and 22 occupation indicators, and some other characteristics.

Here we illustrate the ideas sing a data set of 12697 observations from the March Current Population Survey Supplement 2015.

In this data set, the outcome observations Y_i's are log wage of never-married workers living in the U.S; and the features Z_i's consist of a variety of worker characteristics, including experience, race, education, 23 industry and 22 occupation indicators, and some other characteristics.

Setting

We will estimate the two sets of prediction rules: Linear and Nonlinear Models. In linear models, where we estimate the prediction rule of the form $\hat{g}(Z) = \hat{\beta}^{r} X$, we generate X in two ways:

- in a basic model, X consists of 72 raw regressors Z and a constant;
- in a flexible model, X consists of Z, four polynomials in experience, and all two-way interactions of these variables; this gives us 2336 regressors;

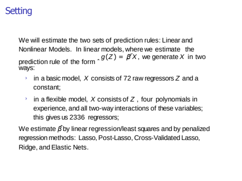
We estimate $\hat{\beta}$ by linear regression/least squares and by penalized regression methods: Lasso, Post-Lasso, Cross-Validated Lasso, Ridge, and Elastic Nets.

2016-07-25

Regression 3.3

└─Assessment of Prediction Quality for for Modern Regression Methods

└─Setting



Setting

We will estimate the two sets of prediction rules: Linear and Nonlinear Models. In linear models, where we estimate the prediction rule of the form $\hat{g}(Z) = \hat{\beta}'X$, we generate X in two ways:

- in a basic model, X consists of 72 raw regressors Z and a constant;
- in a flexible model, X consists of Z, four polynomials in experience, and all two-way interactions of these variables; this gives us 2336 regressors;

We estimate $\hat{\beta}$ by linear regression/least squares and by penalized regression methods: Lasso, Post-Lasso, Cross-Validated Lasso, Ridge, and Elastic Nets.

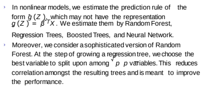
We will estimate or train two sets of prediction rules: Linear and Nonlinear Models

In linear models, we estimate the prediction rule of the form $\hat{g}(Z) = \hat{\beta}'X$, with X generated in two ways:

In the basic model, X consist of 72 raw regressors Z and a constant; In the flexible model, X consist of 2336 constructed regressors, which include the raw regressors Z, four polynomials in experience and all two-way interactions of these regressors.

We estimate $\hat{\beta}$ by linear regression/least squares and by penalized regression methods: Lasso, Post-Lasso, Cross-Validated Lasso, Ridge, and Elastic Nets.

- In nonlinear models, we estimate the prediction rule of the form $\hat{g}(Z)$, which may not have the representation $g(Z) = \hat{\beta}^T X$. We estimate them by Random Forest, Regression Trees, Boosted Trees, and Neural Network.
- Moreover, we consider a sophisticated version of Random Forest. At the step of growing a regression tree, we choose the best variable to split upon among $\sqrt{p}$ p variables. This reduces correlation amongst the resulting trees and is meant to improve the performance.



- In nonlinear models, we estimate the prediction rule of the form $\hat{g}(Z)$, which may not have the representation $g(Z) = \hat{\beta}^T X$. We estimate them by Random Forest, Regression Trees, Boosted Trees, and Neural Network.
- Moreover, we consider a sophisticated version of Random Forest. At the step of growing a regression tree, we choose the best variable to split upon among $\sqrt{p}$ p variables. This reduces correlation amongst the resulting trees and is meant to improve the performance.

In nonlinear models, we estimate the prediction rule of the form $\hat{g}(Z)$. We estimate them by Random Forest, Regression Trees, Boosted Trees, and Neural Network.

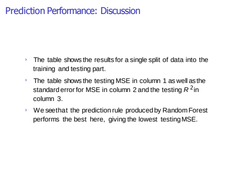
Moreover, here we consider a more sophisticated version of Random Forest, where when growing the regression trees, we choose the best splitting variable among $\sqrt{p}$ p variables. This reduces the correlation amongst resulting trees and is meant to improve performance.

Prediction Performance for the Test/Validation Sample

	MSE	S.E. for MSE	R-squared
Least Squares	0.253	0.017	0.340
Least Squares(Flexible)	11.262	2.915	0.000
Lasso	0.260	0.016	0.322
Lasso(Flexible)	0.259	0.016	0.324
Post-Lasso	0.260	0.017	0.321
Post-Lasso(Flexible)	0.260	0.016	0.321
Cross-Validated lasso	0.273	0.017	0.288
Cross-Validated lasso(Flexible)	0.291	0.017	0.240
Cross-Validated ridge	0.281	0.016	0.266
Cross-Validated ridge(Flexible)	0.285	0.016	0.255
Cross-Validated elnet	0.271	0.017	0.292
Cross-Validated elnet(Flexible)	0.279	0.017	0.271
Random Forest	0.249	0.015	0.350
Boosted Trees	0.259	0.016	0.324
Pruned Tree	0.302	0.017	0.212
Neural Network	0.488	0.045	0.000

Prediction Performance: Discussion

- The table shows the results for a single split of data into the training and testing part.
- The table shows the testing MSE in column 1 as well as the standard error for MSE in column 2 and the testing R^2 in column 3.
- We see that the prediction rule produced by Random Forest performs the best here, giving the lowest testing MSE.



Prediction Performance: Discussion

- The table shows the results for a single split of data into the training and testing part.
- The table shows the testing MSE in column 1 as well as the standard error for MSE in column 2 and the testing R^2 in column 3.
- We see that the prediction rule produced by Random Forest performs the best here, giving the lowest testing MSE.

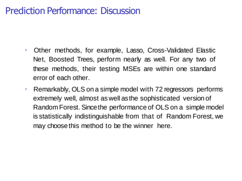
We present the results in the table that you see; we report the results for a single split of data into the training and testing part.

The table shows the testing MSE as well as the standard error associated with it.

We see that the prediction rule produced by Random Forest performs the best here, giving the lowest out-of-sample or test MSE and R^2.

Prediction Performance: Discussion

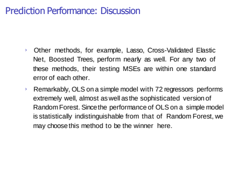
- Other methods, for example, Lasso, Cross-Validated Elastic Net, Boosted Trees, perform nearly as well. For any two of these methods, their testing MSEs are within one standard error of each other.
- Remarkably, OLS on a simple model with 72 regressors performs extremely well, almost as well as the sophisticated version of Random Forest. Since the performance of OLS on a simple model is statistically indistinguishable from that of Random Forest, we may choose this method to be the winner here.



Prediction Performance: Discussion

- Other methods, for example, Lasso, Cross-Validated Elastic Net, Boosted Trees, perform nearly as well. For any two of these methods, their testing MSEs are within one standard error of each other.
- Remarkably, OLS on a simple model with 72 regressors performs extremely well, almost as well as the sophisticated version of Random Forest. Since the performance of OLS on a simple model is statistically indistinguishable from that of Random Forest, we may choose this method to be the winner here.

Other methods, for example, Lasso, Cross-Validated Elastic Net, Boosted Trees, perform nearly as well. They all perform similar, with testing MSE's being within one standard error of each other.
Remarkably, the simple classical OLS on a simple model with 72 regressors performs extremely well, almost as well as the sophisticated version of Random Forest. Since the performance of OLS done on a simple model is statistically indistinguishable from that of Random Forest, we may choose this method to be the winner here.



Prediction Performance: Discussion

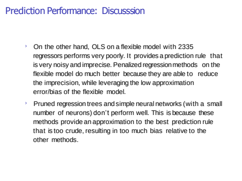
- Other methods, for example, Lasso, Cross-Validated Elastic Net, Boosted Trees, perform nearly as well. For any two of these methods, their testing MSEs are within one standard error of each other.
- Remarkably, OLS on a simple model with 72 regressors performs extremely well, almost as well as the sophisticated version of Random Forest. Since the performance of OLS on a simple model is statistically indistinguishable from that of Random Forest, we may choose this method to be the winner here.

On the other hand, OLS on a flexible model with 2335 regressors performs very poorly. It provides a prediction rule that is very noisy and imprecise. Penalized regression methods do much better because they are able to reduce the imprecision, while leveraging the low approximation error/bias of the flexible model.

Pruned regression trees and simple neural networks (with a small number of neurons) don't perform well. This is because these methods provide an approximation to the best prediction rule that is too crude, resulting in too much bias relative to the other methods.

Prediction Performance: Discusssion

- On the other hand, OLS on a flexible model with 2335 regressors performs very poorly. It provides a prediction rule that is very noisy and imprecise. Penalized regression methods on the flexible model do much better because they are able to reduce the imprecision, while leveraging the low approximation error/bias of the flexible model.
- Pruned regression trees and simple neural networks (with a small number of neurons) don’t perform well. This is because these methods provide an approximation to the best prediction rule that is too crude, resulting in too much bias relative to the other methods.



Prediction Performance: Discusssion

- On the other hand, OLS on a flexible model with 2335 regressors performs very poorly. It provides a prediction rule that is very noisy and imprecise. Penalized regression methods on the flexible model do much better because they are able to reduce the imprecision, while leveraging the low approximation error/bias of the flexible model.
- Pruned regression trees and simple neural networks (with a small number of neurons) don't perform well. This is because these methods provide an approximation to the best prediction rule that is too crude, resulting in too much bias relative to the other methods.

On the other hand classical OLS done a flexible model with 2335 regressors performs very poorly. It provides a prediction rule that is very noisy and imprecise. Penalized regression methods on the flexible model do much better because they are able to reduce the imprecision, while leveraging the low approximation error/bias of the flexible model. Pruned regression trees and simple neural networks (with a small number of neurons) don't perform well. This is because these methods provide an approximation to the best prediction rule that is too crude, resulting in too much bias relative to other methods.

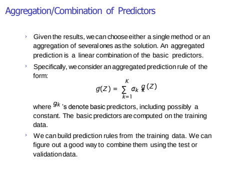
Aggregation/Combination of Predictors

- Given the results, we can choose either a single method or an aggregation of several ones as the solution. An aggregated prediction is a linear combination of the basic predictors.
- Specifically, we consider an aggregated prediction rule of the form:

$$\tilde{g}(Z) = \sum_{k=1}^{K} \tilde{\alpha}_k \hat{g}_k(Z)$$

where $\hat{g}_K$'s denote basic predictors, including possibly a constant. The basic predictors are computed on the training data.

- We can build prediction rules from the training data. We can figure out a good way to combine them using the test or validation data.



Aggregation/Combination of Predictors

- Given the results, we can choose either a single method or an aggregation of several ones as the solution. An aggregated prediction is a linear combination of the basic predictors.
- Specifically, we consider an aggregated prediction rule of the form:

$$\tilde{g}(Z) = \sum_{k=1}^{K} \tilde{\alpha}_k \hat{g}_k(Z)$$

  where g_k 's denote basic predictors, including possibly a constant. The basic predictors are computed on the training data.
- We can build prediction rules from the training data. We can figure out a good way to combine them using the test or validation data.

Given the results presented, we can choose one of the best performing prediction rules. For example, we can select the prediction rules generated by least squares on the simple model, or the prediction rule generated by Lasso on the flexible model, or the prediction rule generated by the Random Forest.

We can also consider aggregations or ensembles of prediction rules, which combine several prediction rules into one.

Specifically, we can consider the aggregated prediction rule of the form:

$$\tilde{g}(Z) = \sum_{k=1}^{K} \tilde{\alpha}_k \hat{g}_k(Z),$$

where $\hat{g}_k$'s denote prediction rules obtained using the training data, including possibly a constant.

We can build prediction rules from the training data. We can figure out a good way to combine them using the test or validation data.

Combining Predictions/Aggregations/Ensemble Learning

- If the number of prediction rules, K , is small, we can figure out the coefficients of the optimal linear combination $\tilde{\alpha}_k$ of the rules using test data V , by simply running least squares of outcomes on the predicted values in the test sample:

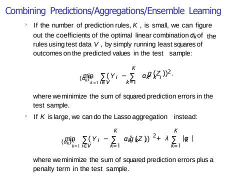
$$\min_{(\alpha_k)_{k=1}^K} \sum_{i \in V} (Y_i - \sum_{k=1}^{K} \alpha_k g_k(Z_i))^2.$$

where we minimize the sum of squared prediction errors in the test sample.

- If K is large, we can do the Lasso aggregation instead:

$$\min_{(\alpha_k)_{k=1}^K} \sum_{i \in V} (Y_i - \sum_{k=1}^{K} \alpha_k g_k(Z_i))^2 + \lambda \sum_{k=1}^{K} |\alpha_k|$$

where we minimize the sum of squared prediction errors plus a penalty term in the test sample.



Combining Predictions/Aggregations/Ensemble Learning

- If the number of prediction rules, K , is small, we can figure out the coefficients of the optimal linear combination $\tilde{\alpha}_k$ of the rules using test data V , by simply running least squares of outcomes on the predicted values in the test sample:

$$\min_{(\alpha_k)_{k=1}^K} \sum_{i\in V} (Y_i - \sum_{k=1}^K \alpha_k g_k(Z_i))^2.$$

where we minimize the sum of squared prediction errors in the test sample.

- If K is large, we can do the Lasso aggregation instead:

$$\min_{(\alpha_k)_{k=1}^K} \sum_{i\in V} (Y_i - \sum_{k=1}^K \alpha_k \hat{g}_k(Z_i))^2 + \lambda \sum_{k=1}^K |\alpha_k|$$

where we minimize the sum of squared prediction errors plus a penalty term in the test sample.

If the number of prediction rules, K , is small, we can figure out the coefficients of the optimal linear combination $\tilde{\alpha}_k$ of the rules using test data V , by simply running least squares of outcomes on the predicted values in the test sample:

$$\min_{(\alpha_k)_{k=1}^K} \sum_{i\in V} (Y_i - \sum_{k=1}^K \alpha_k g_k(Z_i))^2.$$

where we minimize the sum of squared prediction errors in the test sample.

If K is large, we can do Lasso aggregation instead:

$$\min_{(\alpha_k)_{k=1}^K} \sum_{i\in V} (Y_i - \sum_{k=1}^K \alpha_k g_k(Z_i))^2 + \lambda \sum_{k=1}^K |\alpha_k|$$

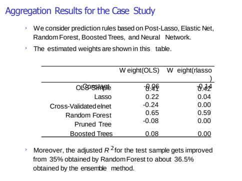
where we minimize the sum of squared prediction errors plus a penalty term in the test sample.

Aggregation Results for the Case Study

- We consider prediction rules based on Post-Lasso, Elastic Net, Random Forest, Boosted Trees, and Neural Network.
- The estimated weights are shown in this table.

	Weight(OLS)	Weight(rlasso)
Constant	-0.06	-0.14
OLS-Simple	0.41	0.42
Lasso	0.22	0.04
Cross-Validated elnet	-0.24	0.00
Random Forest	0.65	0.59
Pruned Tree	-0.08	0.00
Boosted Trees	0.08	0.00

- Moreover, the adjusted R^2 for the test sample gets improved from 35% obtained by Random Forest to about 36.5% obtained by the ensemble method.



Aggregation Results for the Case Study

- We consider prediction rules based on Post-Lasso, Elastic Net, Random Forest, Boosted Trees, and Neural Network.
- The estimated weights are shown in this table.

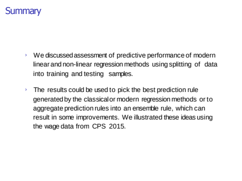
	W eight(OLS)	W eight(rlasso)
Constant	-0.06	-0.14
OLS Simple	0.41	0.42
Lasso	0.22	0.04
Cross-Validated elnet	-0.24	0.00
Random Forest	0.65	0.59
Pruned Tree	-0.08	0.00
Boosted Trees	0.08	0.00

- Moreover, the adjusted R^2 for the test sample gets improved from 35% obtained by Random Forest to about 36.5% obtained by the ensemble method.

We consider prediction rules based on Post-Lasso, Elastic Net, Random Forest, Boosted Trees, and Neural Network.
We estimated the coefficients α_k's using least squares and Lasso. From the estimated coefficients, we see that most of the weight goes to the prediction rules generated by least squares on a simple model and by the Random Forest. Other prediction rules receive considerably less weight.
Moreover, the adjusted R^2 for the test simple gets improved from 35% obtained by Random Forest to about 36.5% obtained by the ensemble method.

Summary

- We discussed assessment of predictive performance of modern linear and non-linear regression methods using splitting of data into training and testing samples.
- The results could be used to pick the best prediction rule generated by the classical or modern regression methods or to aggregate prediction rules into an ensemble rule, which can result in some improvements. We illustrated these ideas using the wage data from CPS 2015.



Summary

- We discussed assessment of predictive performance of modern linear and non-linear regression methods using splitting of data into training and testing samples.
- The results could be used to pick the best prediction rule generated by the classical or modern regression methods or to aggregate prediction rules into an ensemble rule, which can result in some improvements. We illustrated these ideas using the wage data from CPS 2015.

We discussed assessment of predictive performance of modern linear and non-linear regression methods using splitting of data into training and testing samples.

The results could be used to pick the best prediction rule generated by the classical or modern regression methods or to aggregate prediction rules into an ensemble rule, which can result in some improvements. We illustrated these ideas using the wage data from CPS 2015.

Module 2 – Regression and Prediction

THANK YOU

Case Study 4: Predicting Wages 2

PROFESSIONAL EDUCATION
Digital Programs