Model Evaluation

Model Evaluation

Overview

- Once we have a model ready, we need to determine if it meets our goals:
  - Is it accurate enough for your needs?
  - Does it generalize well?
  - Does it perform better than "the obvious guess"?
  - Does it perform better than whatever estimate we currently use?
  - Do the results of the model (coefficients, clusters, rules, confidence intervals, significances, and diagnostics) make sense in the context of the problem domain?
- If the answer is "NO" to any of the above questions, it’s time to loop back to the modeling step
  - Or decide that the data doesn’t support the goal you’re trying to achieve.

Source: Practical Data Science with R - Nina Zumel, John Mount

Model Evaluation

Evaluation – Case Scenario

- In the loan application example, the first thing to check is whether the rules that the model discovered make sense
- Looking at decision tree structure, we don't notice any obviously strange rules, so you can go ahead and evaluate the model's accuracy
- A good summary of classifier accuracy is the confusion matrix, which tabulates actual classifications against predicted ones.
- We create a confusion matrix where rows represent actual loan status, and columns represent predicted loan status
- conf_mat ["GoodLoan", "BadLoan"] refers to the element conf_mat[2, 1]
  - The number of actual good loans that the model predicted were bad
- The diagonal entries of the matrix represent correct predictions.

Source: Practical Data Science with R - Nina Zumel, John Mount

Model Evaluation

Confusion Matrix

- A confusion matrix is a table that is often used to evaluate the performance of a classification model (or "classifier")
- It works on a set of test data for which the *true values are known*
- There are two possible predicted classes: "YES" and "NO"
- If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.
- The classifier made a total of 165 predictions
  - E.g., 165 patients were being tested for the presence of that disease
- Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
- In reality, 105 patients in the sample have the disease, and 60 patients do not.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Source: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology

Model Evaluation

Confusion Matrix

- True positives (TP):
  - These are cases in which the model predicted yes (they have the disease), and the patients actually do have the disease.
- True negatives (TN):
  - The model predicted no, and they don't have the disease.
- False positives (FP):
  - The model predicted YES, but they don't actually have the disease. (Also known as a "Type I error.")
- False negatives (FN):
  - The model predicted NO, but they actually do have the disease. (Also known as a "Type II error.")

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Source: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology

Model Evaluation

Confusion Matrix

Term	Description	Calculation
Accuracy	Overall, how often is the classifier correct?	• (TP+TN)/total = (100+50)/165 = 0.91
Misclassification Rate	Overall, how often is it wrong?	• (FP+FN)/total = (10+5)/165 = 0.09 • Equivalent to 1 minus Accuracy • Also known as "Error Rate"
True Positive Rate (Sensitivity or Recall)	When it's actually YES, how often does it predict YES?	• TP/actual YES = 100/105 = 0.95
False Positive Rate	When it's actually NO, how often does it predict YES?	• FP/actual NO = 10/60 = 0.17
True Negative Rate: (Specificity)	When it's actually NO, how often does it predict NO?	• TN/actual NO = 50/60 = 0.83 • Equivalent to 1 minus False Positive Rate
Precision	When it predicts YES, how often is it correct?	• TP/predicted YES = 100/110 = 0.91
Prevalence	How often does the YES condition actually occur in our sample?	• Actual YES/total = 105/165 = 0.64

Source: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology

Model Evaluation

innovate achieve lead

Evaluation – Case Scenario

How to install all the packages needed to run examples in the book can be found here: https://github.com/WinVector/PDSwR2/blob/master/packages.R.

This file can be found at https://github.com/WinVector/PDSwR2/tree/master/Statlog.

```
library("rpart")
load("loan_model_example.RData")
conf_mat <-
     table(actual = d$Loan_status, pred = predict(model, type = 'class'))

##            pred
## actual      BadLoan GoodLoan
##   BadLoan        41      259
##   GoodLoan       13      687

(accuracy <- sum(diag(conf_mat)) / sum(conf_mat))
## [1] 0.728
(precision <- conf_mat["BadLoan", "BadLoan"] / sum(conf_mat[, "BadLoan"])
## [1] 0.7592593
(recall <- conf_mat["BadLoan", "BadLoan"] / sum(conf_mat["BadLoan", ]))
## [1] 0.1366667

(fpr <- conf_mat["GoodLoan","BadLoan"] / sum(conf_mat["GoodLoan", ]))
## [1] 0.01857143
```

Creates the confusion matrix

Overall model accuracy: 73% of the predictions were correct.

False positive rate: 2% of the good applicants were mistakenly identified as bad.

Model precision: 76% of the applicants predicted as bad really did default.

Model recall: the model found 14% of the defaulting loans.

Source: Practical Data Science with R - Nina Zumel, John Mount

Model Evaluation

innovate achieve lead

Evaluation – Case Scenario

Term	Calculation
Accuracy	• (TP+TN)/total = (41+687)/1000 = 0.728
Misclassification Rate	• (FP+FN)/total = (13+259)/1000 = 0.272 • Equivalent to 1 minus Accuracy • Also known as "Error Rate"
True Positive Rate	• TP/actual yes = 41/300 = 0.137 • Also known as "Sensitivity" or "Recall"
False Positive Rate	• FP/actual no = 13/700 = 0.01857
True Negative Rate:	• TN/actual no = 687/700 = 0.98143 • Equivalent to 1 minus False Positive Rate • Also known as "Specificity"
Precision	• TP/predicted yes = 41/54 = 0.759
Prevalence	• Actual yes/total = 300/1000 = 0.64

		Predicted		
	n = 1000	Bad Loan Yes	Good Loan No	
Actual	Bad Loan Yes	TP = 41	FN = 259	300
	Good Loan No	FP = 13	TN = 687	700
		54	946	

Our prediction is for Bad Loans. So,

- Yes means, Yes, it's a bad loan
- No means, No, it's not a bad loan

Source: Practical Data Science with R - Nina Zumel, John Mount

Model Presentation

Model Presentation

Overview

- This phase involves presenting our results to stakeholders
- We also document the model for those in the organization who are responsible for using, running, and maintaining the model once it has been deployed
- Different audiences require different kinds of information
- Business-oriented audiences want to understand the impact of our findings in terms of business metrics

Model Presentation

Case Scenario

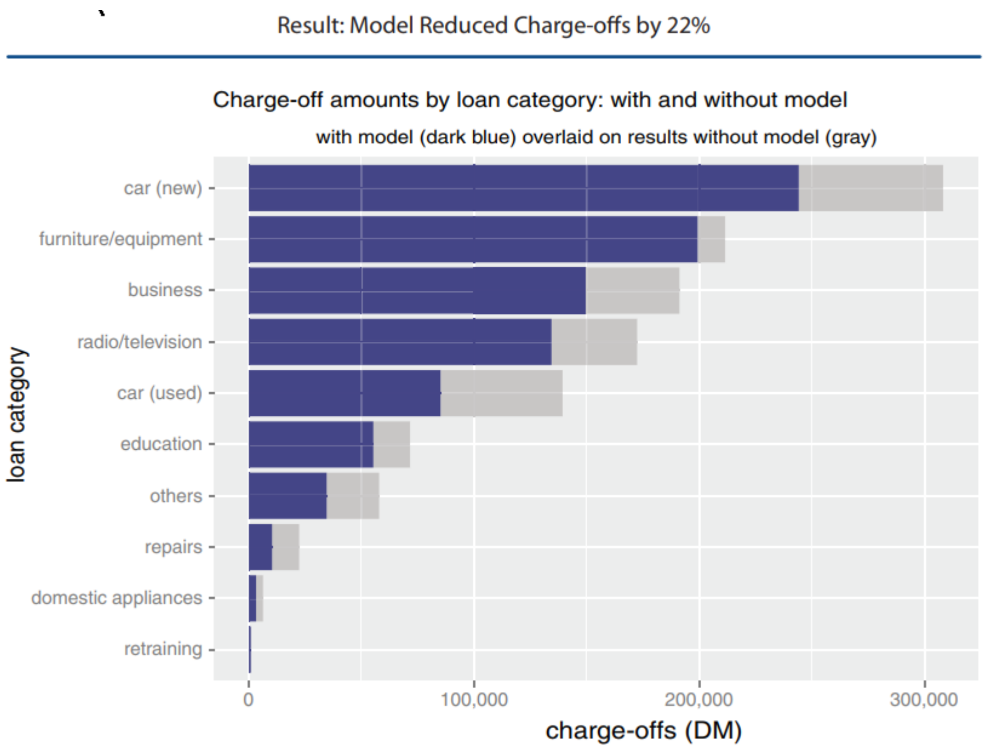
- In the loan example, our business audience is interested in knowing how our model can reduce chargeoffs
  - Chargeoff = the money that the bank loses to bad loans
- Suppose our model identified a set of bad loans that amounted to 22% of the total money lost to defaults
- Then our presentation should emphasize that the model can potentially reduce the bank's losses by that amount
  - See figure in the next slide
- We may also give other interesting findings or recommendations, such as:
  - New car loans are much riskier than used car loans
  - Most losses are tied to bad car loans and bad equipment loans (assuming that the audience didn't already know these facts)

Source: Practical Data Science with R - Nina Zumel, John Mount

Model Presentation

Case Scenario

- Our presentation for the loan officers should emphasize:
  - How should they interpret the model?
  - What does the model output look like?
  - If the model provides a trace of which rules in the decision tree executed
    - How do they read that?
  - If the model provides a confidence score in addition to a classification
    - How should they use the confidence score?
  - When might they potentially overrule the model?



Source: Practical Data Science with R - Nina Zumel, John Mount

Model Deployment

Concept

- Finally, the model is put into operation
- In many organizations, the data scientist no longer has primary responsibility for the day-to-day operation of the model
- But you still should ensure that the model will run smoothly and won't make disastrous decisions
- You also want to make sure that the model can be updated as its environment changes
- And in many situations, the model will initially be deployed in a small pilot program
- The test might bring out issues that you didn't anticipate, and you may have to adjust the model accordingly

Source: Practical Data Science with R - Nina Zumel, John Mount

Model Deployment

Case Scenario

- When we deploy the model, we might find that loan officers frequently override the model in certain situations because it contradicts their intuition
- Is their intuition wrong? Or
- Is your model incomplete? Or
- More positively, your model may perform so successfully that the bank wants you to extend it to home loans as well

Thank You!