

Model Evaluation

Matthew McDonald

Regression vs Classification

- Definition:
 - **Regression:** Predicts continuous outcomes. It estimates a mapping function (f) from input variables (X) to a continuous output variable (Y).
 - **Classification:** Predicts discrete outcomes. It categorizes input data into two or more classes.
- Output Type:
 - **Regression:** The output is a real or continuous value (e.g., salary, price).
 - **Classification:** The output is a category (e.g., default or no default).
- Both regression and classification are types of supervised machine learning algorithms, where a model is trained according to the existing model along with correctly labeled data

RMSE

- Definition: The square root of the average of the squared differences between the predicted and actual values. It measures the standard deviation of the residuals (prediction errors).
- Interpretation: Represents the average error made by the model in predicting the outcome. Lower values indicate better fit.
- Scale: Depends on the target variable's scale. Higher RMSE values may not necessarily indicate a poor model, especially if the target variable has a wide range.
- Sensitivity: More sensitive to outliers than other metrics, such as MAE (Mean Absolute Error).

RMSE Formula

- $\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
- n is the number of observations.
- y_i is the actual value of the observation.
- \hat{y}_i is the predicted value.
- The sum $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ calculates the squared differences between the actual and predicted values.

R^2

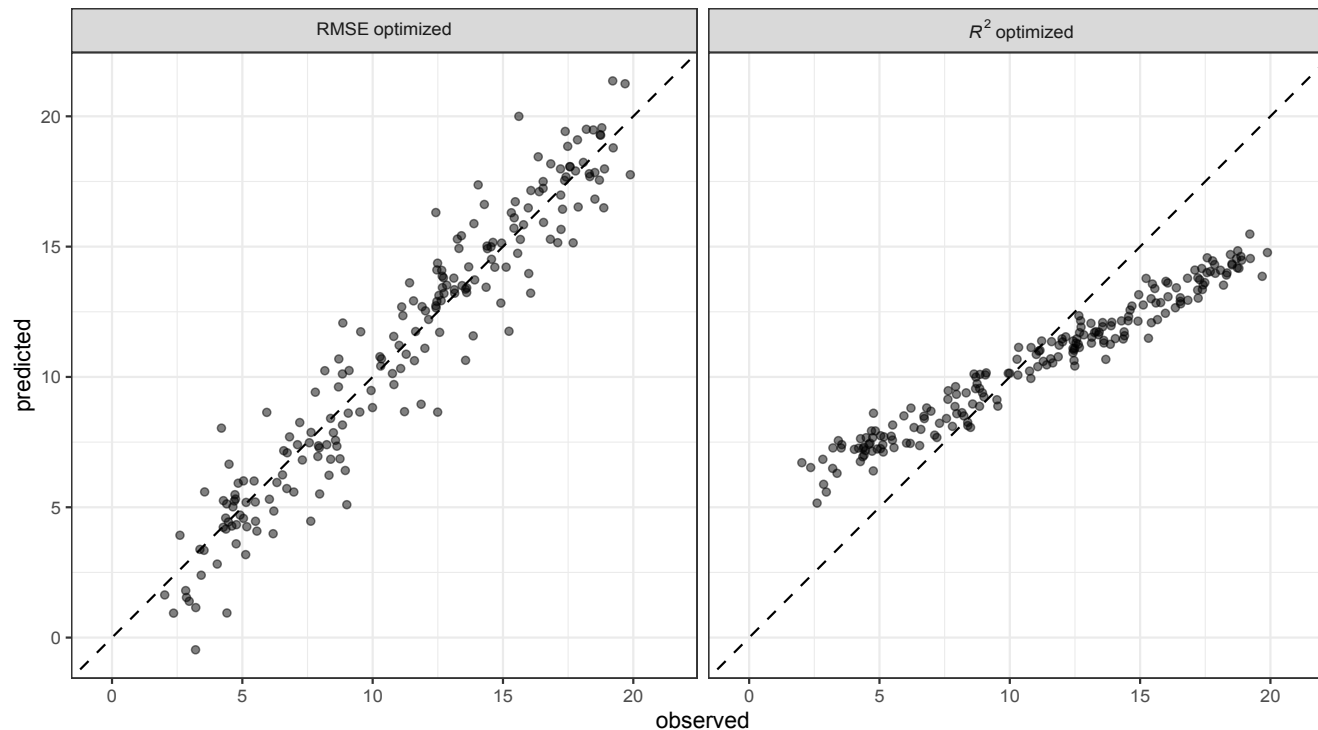
- Definition: The proportion of the variance in the dependent variable that is predictable from the independent variables. It is a statistical measure of how close the data are to the fitted regression line.
- Interpretation: Values range from 0 to 1. A higher value indicates a better fit, with 1 meaning the model explains all the variability of the response data around its mean.
- Scale-Independent: Unlike RMSE, R-squared is a normalized measure, making it easier to compare the goodness of fit across different datasets and models.
- Limitation: Can be misleadingly high in models with many predictors or when using higher-order polynomials. Adjusted R-squared is often used to account for the number of predictors.

R^2 Formula

- $R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$
- n , y_i and \hat{y}_i are the defined the same as in *RMSE*
- The numerator, $\sum_{i=1}^n (y_i - \hat{y}_i)^2$, is the sum of the squared differences between the actual and the predicted values (also known as the sum of squares of residuals).
- The denominator, $\sum_{i=1}^n (y_i - \bar{y})^2$, is the total sum of squares (TSS) or the sum of the squared differences between the actual values and the mean of the actual values.

RMSE vs R^2

RMSE measures *accuracy* while R^2 measures *correlation*.



Observed versus predicted values for models that are optimized using the RMSE compared to the coefficient of determination

Calculating RMSE using yardstick




```
1 library (yardstick)
2
3 rmse(data_with_predictions, truth = latency, estimate = .pred)
```

```
#> # A tibble: 1 × 3
#>   .metric .estimator .estimate
#>   <chr>    <chr>      <dbl>
#> 1 rmse     standard      59.2
```


Calculating Multiple Metrics

```
1 metrics <- metric_set(rmse, rsq, mae)
2 metrics(data_with_predictions, truth = latency, estimate = .pred)
```

```
#> # A tibble: 3 × 3
#>   .metric .estimator .estimate
#>   <chr>    <chr>      <dbl>
#> 1 rmse    standard      59.2
#> 2 rsq     standard       0.380
#> 3 mae     standard       40.2
```

- RMSE: difference between the predicted and observed values 
- R^2 : squared correlation between the predicted and observed values 
- MAE: similar to RMSE, but mean absolute error 

Classification Metrics

Definitions:

- **Hard Predictions:** provide the final classification result directly, indicating the class to which the input data is most likely to belong, without showing any uncertainty or probability of the decision.
 - Example: A model predicts an obligor is either “defaulter” (1) or “not a defualter” (0), with no indication of how likely it is to be a defaulter.
- **Soft Predictions:** provide a probability distribution over all possible classes, indicating the likelihood that the input data belongs to each class.
 - Example: A model predicts an obligor has a 90% probability of being a “defaulter” and a 10% probability of being “not a defaulter”.

Confusion Matrix

For Hard Predictions, we can create a confusion matrix

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Components of Confusion Matrix

1. **True Positives (TP):** Instances correctly predicted as positive.
2. **True Negatives (TN):** Instances correctly predicted as negative.
3. **False Positives (FP):** Instances incorrectly predicted as positive, also known as Type I error.
4. **False Negatives (FN):** Instances incorrectly predicted as negative, also known as Type II error.

Derived Metrics from Confusion Matrix

From the confusion matrix, several important metrics can be calculated, including:

- **Accuracy:** Overall correctness of the model. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- **Precision:** Proportion of positive identifications that were actually correct.
 $Precision = \frac{TP}{TP+FP}$
- **Recall (Sensitivity or True Positive Rate):** Proportion of actual positives that were identified correctly. $Recall = \frac{TP}{TP+FN}$
- **Specificity (True Negative Rate):** Proportion of actual negatives that were identified correctly. $Specificity = \frac{TN}{TN+FP}$
- **F1 Score:** Harmonic mean of precision and recall, providing a single metric to assess the balance between them. $F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$

Mapping Soft Predictions into Hard Predictions

For two classes, the customary cutoff to map probabilities into predictions is 50%. If the probability of class 1 is $\geq 50\%$, they would be labelled as *Class1*

What happens if you change the cutoff?

- Increasing makes it harder to be called *Class1* \implies fewer predicted events, specificity \uparrow , sensitivity \downarrow
- Decreasing makes it harder to be called *Class1* \implies fewer predicted events, specificity \uparrow , sensitivity \downarrow

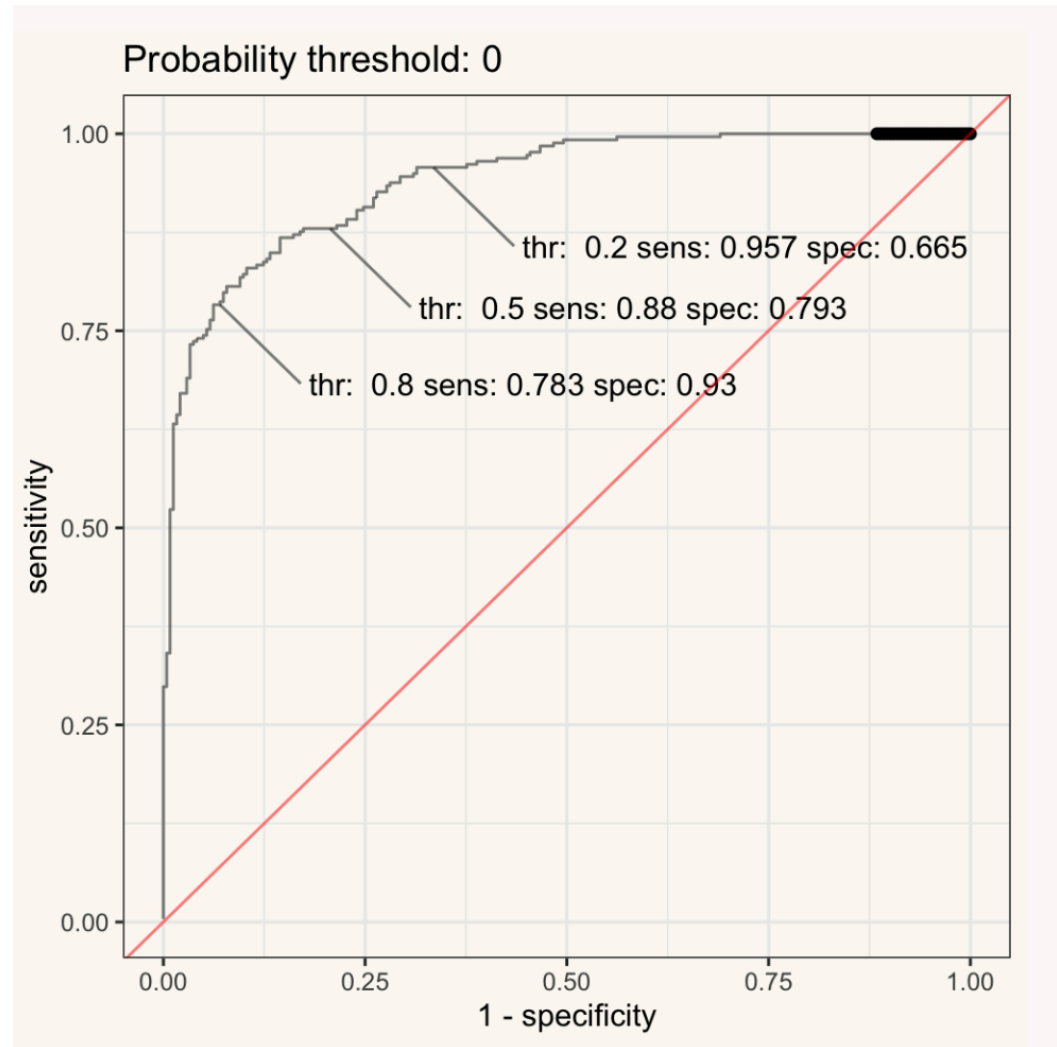
With two classes, the **Receiver Operating Characteristic (ROC) curve** can be used to estimate performance using a combination of sensitivity and specificity.

To create the curve, many alternative cutoffs are evaluated.

For each cutoff, we calculate the sensitivity and specificity.

The ROC curve plots the sensitivity (True Positive Rate) versus $1 - \text{specificity}$ (False Positive Rate)

The ROC Curve



AUC

Integral Relationship: The AUC (Area Under the Curve) represents the integral or the total area under the ROC (Receiver Operating Characteristic) curve. It quantifies the overall ability of the model to discriminate between positive and negative classes across all possible thresholds, with a higher AUC indicating better model performance and discrimination capability.

Classification Metrics with yardstick

```
1 data(two_class_example)
2 tibble(two_class_example)
```

```
#> # A tibble: 500 × 4
#>   truth    Class1    Class2 predicted
#>   <fct>    <dbl>    <dbl> <fct>
#> 1 Class2 0.00359 0.996   Class2
#> 2 Class1 0.679    0.321   Class1
#> 3 Class2 0.111    0.889   Class2
#> 4 Class1 0.735    0.265   Class1
#> 5 Class2 0.0162   0.984   Class2
#> 6 Class1 0.999    0.000725 Class1
#> 7 Class1 0.999    0.000799 Class1
#> 8 Class1 0.812    0.188   Class1
#> 9 Class2 0.457    0.543   Class2
#> 10 Class2 0.0976   0.902   Class2
#> # i 490 more rows
```

Hard Prediction Metrics

```
1 # A confusion matrix:
2 conf_mat(two_class_example,
3           truth = truth,
4           estimate = predicted)
```

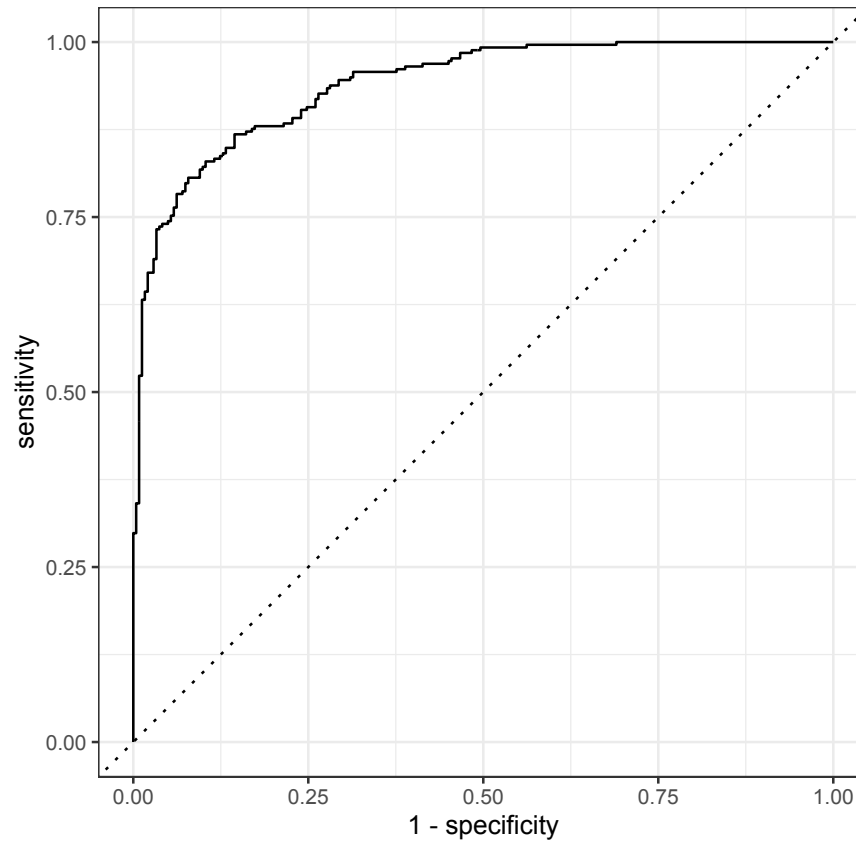
```
#>           Truth
#> Prediction Class1 Class2
#>      Class1      227      50
#>      Class2       31     192
```

```
1 cm <- metric_set(accuracy,
2                  sens,
3                  spec)
4 cm(two_class_example,
5     truth = truth,
6     estimate = predicted)
```

```
#> # A tibble: 3 × 3
#>   .metric .estimator .estimate
#>   <chr>    <chr>      <dbl>
#> 1 accuracy binary      0.838
#> 2 sens     binary      0.880
#> 3 spec     binary      0.793
```

ROC Curve

```
1 two_class_curve <- roc_curve(two_class_example, truth, Class1)
2 two_class_curve %>% autoplot()
```



Calculating AUC

```
1 roc_auc(two_class_example, truth, Class1)
```

```
#> # A tibble: 1 × 3  
#>   .metric .estimator .estimate  
#>   <chr>   <chr>       <dbl>  
#> 1 roc_auc binary      0.939
```