Basic Principles: Exercises

Dr. José Ramón Iglesias
DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingenieria Electronica
Universidad Popular del Cesar

Exercise 1
Evaluation Metrics

Evaluation Metrics

1. *Explain the meaning of the following terms:*
   - *True positive, true negative, false positive, false negative*
   - *Sensitivity and specificity*
   - *Precision and recall*
   - *Overall accuracy and Cohen's kappa.*
2. *When is it preferable to use the overall accuracy? When the Cohen's kappa?*

Explain the meaning of the following terms:

- *True positive, true negative, false positive, false negative*

All terms are based on a two-class decision problem. Arbitrarily, take one class as the **POSITIVE** class and the remaining class as the **NEGATIVE** class.

TRUE POSITIVES (**TP**): Original class is **positive** and predictor class is also **positive** (correct)

TRUE NEGATIVES (**TN**): Original class is **negative** and predictor class is also **negative** (correct)

FALSE NEGATIVES (**FN**): Original class is **positive** and predictor class is **negative** (wrong)

FALSE POSITIVES (**FP**): Original class is **negative** and predictor class is **positive** (wrong)

Correct predictions

True class	Predicted class	
	POSITIVE	**NEGATIVE**
POSITIVE	TRUE POSITIVES	FALSE NEGATIVES
NEGATIVE	FALSE POSITIVES	TRUE NEGATIVES

Explain the meaning of the following terms:

- *Sensitivity and specificity*

- Sensitivity vs. Specificity

$$sensitivity = \frac{TP}{TP+FN}$$

Fraction of **positive** events correctly classified

$$specificity = \frac{TN}{TN+FP}$$

Fraction of **negative** events correctly classified

Both measures must be high!

Explain the meaning of the following terms:

- *Precision and recall*

- Precision vs. Recall

$$recall = \frac{TP}{TP + FN}$$

Same as sensitivity

$$precision = \frac{TP}{TP + FP}$$

Fraction of predicted positives that are actually positive

- F-Measure

$$F = 2\frac{precision \cdot recall}{precision + recall}$$

Explain the meaning of the following terms:

- *Overall accuracy and Cohen's kappa.*

- Overall Accuracy

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{n}$$

Fraction of correctly classified data

- Cohen‘s Kappa

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

Given the a priori probability of the class and given the probability of the model to choose the class, how much better is overall accuracy with respect to just following the pure probabilities

2. Overall accuracy vs. Cohen‘s kappa

When is it preferable to use the overall accuracy? When the Cohen's kappa?

- The more the classes in the dataset are unevenly distributed, the more it is advisable to use Cohen‘s kappa instead of Overall Accuracy.

- An alternative is to artificially produce a dataset with equally distributed classes and calculate the Overall Accuracy for it.

Exercise 2
ROC Curve

ROC curve

- *The adult data set was classified using two models (see the classification results in the image below).*

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

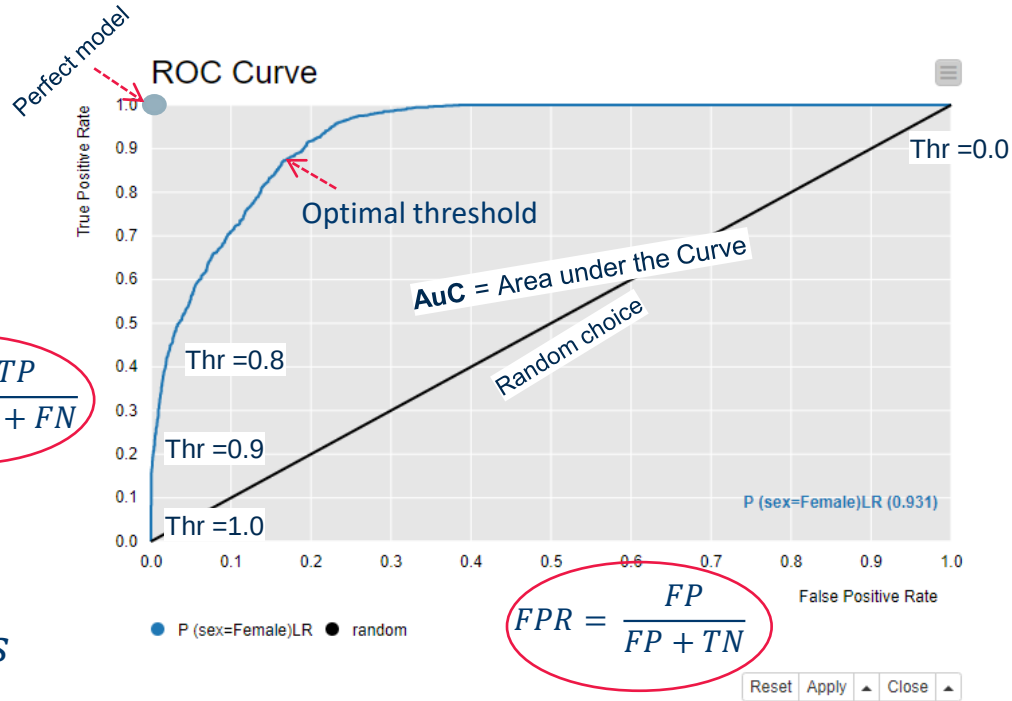
- *The correct classification can be found in the attribute income. The columns P(income=<= 50K)M1 and P(income=<= 50K)M2 contain the probability that this person belongs to the positive class <= 50K according to the prediction of models M1 and M2, respectively.*
  a. *Draw in one diagram (by hand) the ROC curve for the two models and explain how the ROC curve has been drawn.*
  b. *Where would the perfect curve be, where is the random guess, and which area under the curve is supposed to be worse than random?*
  c. *Which of the algorithms better classified the data and why?*

Classification model performance as reported by:

True classs	Predicted class	
	POSITIVE	**NEGATIVE**
POSITIVE	TRUE POSITIVES	FALSE NEGATIVES
NEGATIVE	FALSE POSITIVES	TRUE NEGATIVES

- False positive rate (FPR)
  - negative events **in**correctly classified as positive
- True positive rate (TPR)
  - positive events correctly classified as positive

$$TPR = \frac{TP}{TP + FN}$$



$$FPR = \frac{FP}{FP + TN}$$

$$P(pos|x) > threshold \Rightarrow class\ pos$$

Threshold = 1.0

Positive event = <=50K - Threshold = 1.0 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	0	7
Negative	0	3

$$TPR = \frac{0}{0+7} = 0.00$$

$$FPR = \frac{0}{0+3} = 0.00$$

Model 2

	Positive	Negative
Positive	0	7
Negative	0	3

$$TPR = \frac{0}{0+7} = 0.00$$

$$FPR = \frac{0}{0+3} = 0.00$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.9

Positive event = <=50K - Threshold = 0.9 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	1	6
Negative	0	3

$$TPR = \frac{1}{1+6} = 0.14$$

$$FPR = \frac{0}{0+3} = 0.00$$

Model 2

	Positive	Negative
Positive	1	6
Negative	0	3

$$TPR = \frac{1}{1+6} = 0.14$$

$$FPR = \frac{0}{0+3} = 0.00$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.8

Positive event = <=50K - Threshold = 0.8 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	1	6
Negative	0	3

$$TPR = \frac{1}{1+6} = 0.14$$

$$FPR = \frac{0}{0+3} = 0.00$$

Model 2

	Positive	Negative
Positive	1	6
Negative	0	3

$$TPR = \frac{1}{1+6} = 0.14$$

$$FPR = \frac{0}{0+3} = 0.00$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.7

Positive event = <=50K - Threshold = 0.7 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	5	2
Negative	0	3

$$TPR = \frac{5}{5+2} = 0.71$$

$$FPR = \frac{0}{0+3} = 0.00$$

Model 2

	Positive	Negative
Positive	2	5
Negative	0	3

$$TPR = \frac{2}{2+5} = 0.28$$

$$FPR = \frac{0}{0+3} = 0.00$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.6

Positive event = <=50K - Threshold = 0.6 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	5	2
Negative	1	2

$$TPR = \frac{5}{5+2} = 0.71$$

$$FPR = \frac{1}{1+2} = 0.33$$

Model 2

	Positive	Negative
Positive	3	4
Negative	2	3

$$TPR = \frac{3}{3+4} = 0.42$$

$$FPR = \frac{2}{2+1} = 0.66$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.5

Positive event = <=50K - Threshold = 0.5 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	5	2
Negative	1	2

$$TPR = \frac{5}{5+2} = 0.71$$

$$FPR = \frac{1}{1+2} = 0.33$$

Model 2

	Positive	Negative
Positive	3	4
Negative	2	3

$$TPR = \frac{3}{3+4} = 0.42$$

$$FPR = \frac{2}{2+1} = 0.66$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.4

Positive event = <=50K - Threshold = 0.4 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	6	1
Negative	1	2

$$TPR = \frac{6}{6+1} = 0.86$$

$$FPR = \frac{1}{1+2} = 0.33$$

Model 2

	Positive	Negative
Positive	4	3
Negative	2	3

$$TPR = \frac{4}{4+3} = 0.57$$

$$FPR = \frac{2}{2+1} = 0.66$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	income	P (income=<=50K)M1	Prediction (income)M1	P (income=<=50K)M2	Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.3

Positive event = <=50K - Threshold = 0.3 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	6	1
Negative	1	2

$$TPR = \frac{6}{6+1} = 0.86$$

$$FPR = \frac{1}{1+2} = 0.33$$

Model 2

	Positive	Negative
Positive	6	1
Negative	3	0

$$TPR = \frac{6}{6+1} = 0.86$$

$$FPR = \frac{3}{3+0} = 1.00$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.2

Positive event = <=50K - Threshold = 0.2 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	6	1
Negative	2	1

$$TPR = \frac{6}{6+1} = 0.86$$

$$FPR = \frac{2}{2+1} = 0.66$$

Model 2

	Positive	Negative
Positive	7	0
Negative	3	0

$$TPR = \frac{7}{7+0} = 1.00$$

$$FPR = \frac{3}{3+0} = 1.00$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	income	P (income=<=50K)M1	Prediction (income)M1	P (income=<=50K)M2	Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.1

Positive event = <=50K - Threshold = 0.1 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	6	1
Negative	2	1

$$TPR = \frac{7}{7+0} = 1.00$$

$$FPR = \frac{3}{3+0} = 1.00$$

Model 2

	Positive	Negative
Positive	7	0
Negative	3	0

$$TPR = \frac{7}{7+0} = 1.00$$

$$FPR = \frac{3}{3+0} = 1.00$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

Threshold = 0.0

Positive event = <=50K - Threshold = 0.0 - If P (pos. Event) > threshold => positive

Model 1

	Positive	Negative
Positive	6	1
Negative	2	1

$$TPR = \frac{7}{7+0} = 1.00$$
$$FPR = \frac{3}{3+0} = 1.00$$

Model 2

	Positive	Negative
Positive	7	0
Negative	3	0

$$TPR = \frac{7}{7+0} = 1.00$$
$$FPR = \frac{3}{3+0} = 1.00$$

$$TPR = \frac{TP}{TP + FN}$$
$$FPR = \frac{FP}{FP + TN}$$

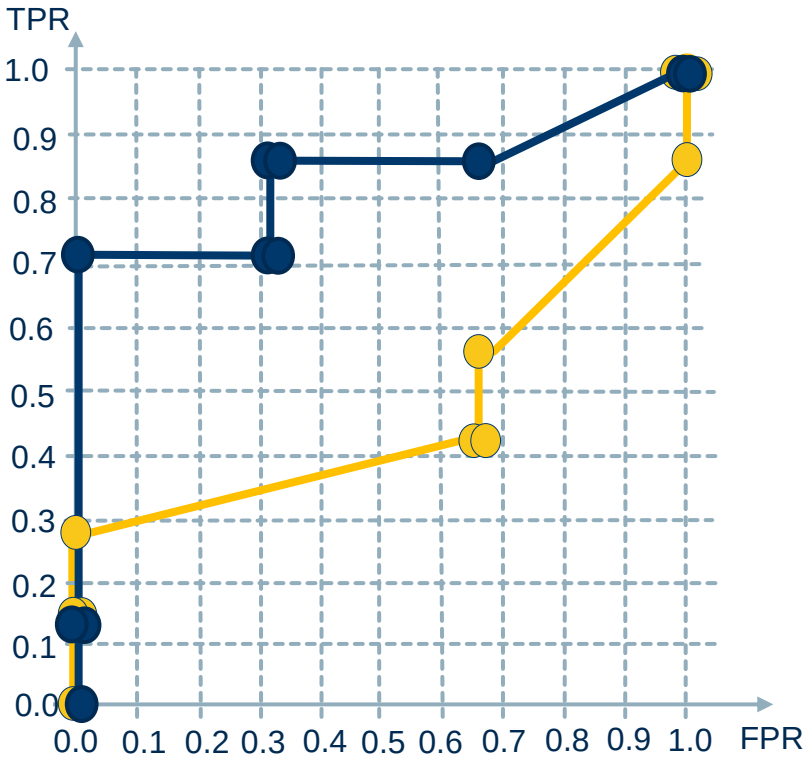
Table "adult_classified.csv" - Rows: 10 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S income	D P (income=<=50K)M1	S Prediction (income)M1	D P (income=<=50K)M2	S Prediction (income)M2
Row0	<=50K	0.15	>50K	0.21	>50K
Row1	>50K	0.25	>50K	0.37	>50K
Row2	<=50K	0.75	<=50K	0.39	>50K
Row3	>50K	0.2	>50K	0.62	<=50K
Row4	<=50K	0.75	<=50K	0.93	<=50K
Row5	<=50K	0.72	<=50K	0.78	<=50K
Row6	<=50K	0.5	<=50K	0.45	>50K
Row7	>50K	0.67	<=50K	0.66	<=50K
Row8	<=50K	0.75	<=50K	0.69	<=50K
Row9	<=50K	1	<=50K	0.35	>50K

ROC Curve

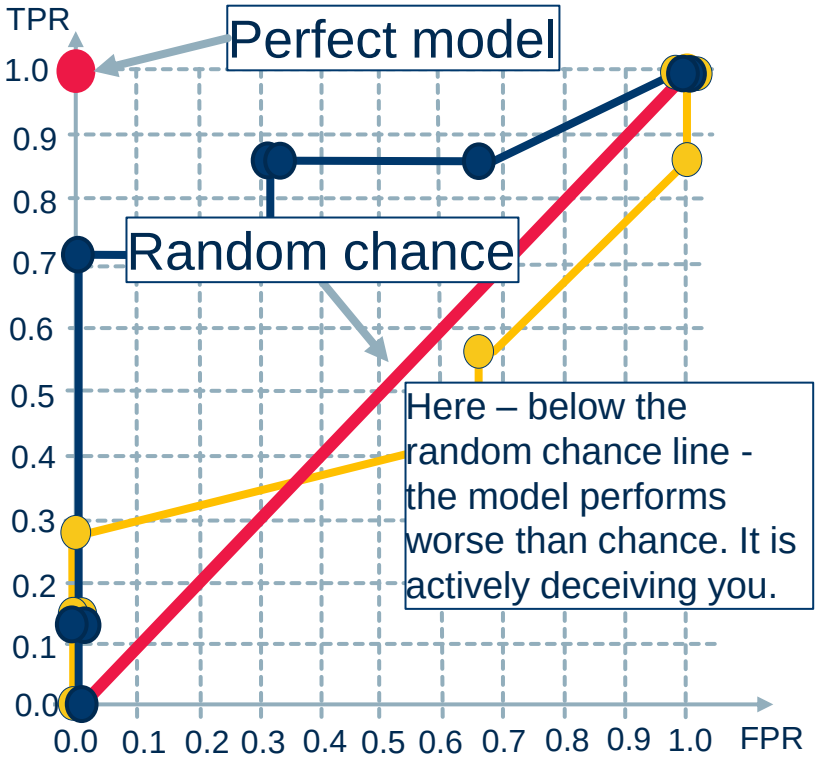
a) Draw in one diagram (by hand) the ROC curve for the two models and explain how the ROC curve has been drawn

	Model 1		Model 2	
threshold	TPR	FPR	TPR	FPR
1.0	0.00	0.00	0.00	0.00
0.9	0.14	0.00	0.14	0.00
0.8	0.14	0.00	0.14	0.00
0.7	0.71	0.00	0.28	0.00
0.6	0.71	0.33	0.42	0.66
0.5	0.71	0.33	0.42	0.66
0.4	0.86	0.33	0.57	0.66
0.3	0.86	0.33	0.86	1.00
0.2	0.86	0.66	1.00	1.00
0.1	1.00	1.00	1.00	1.00
0.0	1.00	1.00	1.00	1.00



b) Where would the perfect curve be, where is the random guess, and which area under the curve is supposed to be worse than random?

	Model 1		Model 2	
threshold	TPR	FPR	TPR	FPR
1.0	0.00	0.00	0.00	0.00
0.9	0.14	0.00	0.14	0.00
0.8	0.14	0.00	0.14	0.00
0.7	0.71	0.00	0.28	0.00
0.6	0.71	0.33	0.42	0.66
0.5	0.71	0.33	0.42	0.66
0.4	0.86	0.33	0.57	0.66
0.3	0.86	0.33	0.86	1.00
0.2	0.86	0.66	1.00	1.00
0.1	1.00	1.00	1.00	1.00
0.0	1.00	1.00	1.00	1.00

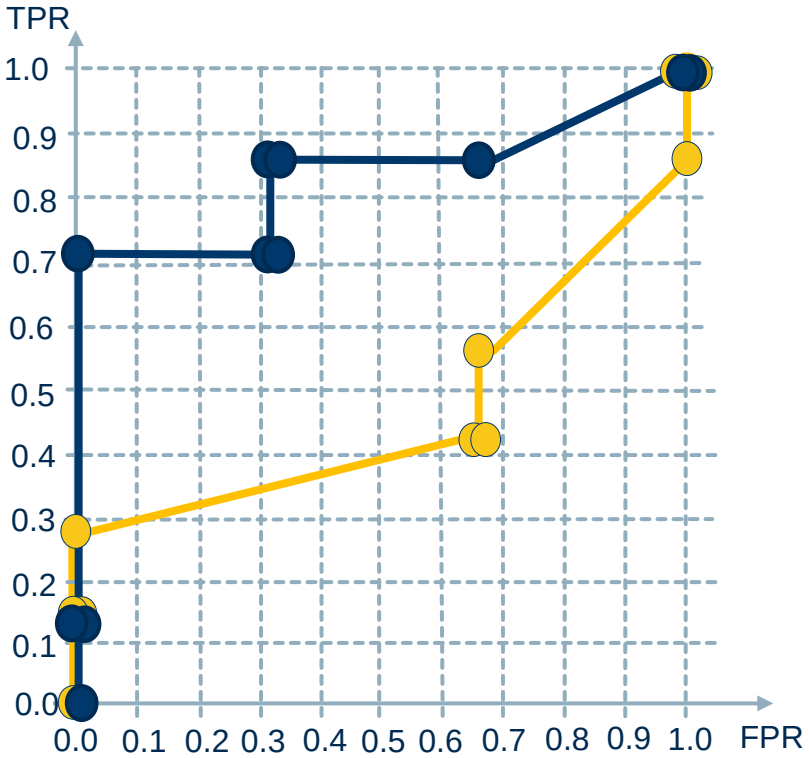


c) Which of the algorithms better classified the data and why?

	Model 1	Model 2

Model 1 (blue line) performs better than model 2 (yellow line), since the ROC curve of model 1 lies above the ROC curve of model 2.

As a consequence the Area under the Curve (AuC) is bigger for model 1 (blue line) than for model 2 (yellow line). AuC can be considered an indicator of the classifier quality.



Exercise 3
Gradient Descent

– *Explain how you can minimize the following formula using gradient descent:*

$$f(a,b) = a^2b - b^3 - 11a + 12b^2 + 15$$

a) *List all constraints that a programmer needs to find values (a,b) that minimize f(a,b).*

b) *Choose a random starting point e.g. (0.0) and apply the algorithm for 5 iterations*

Gradient Descent

Explain how you can minimize the following formula using gradient descent:

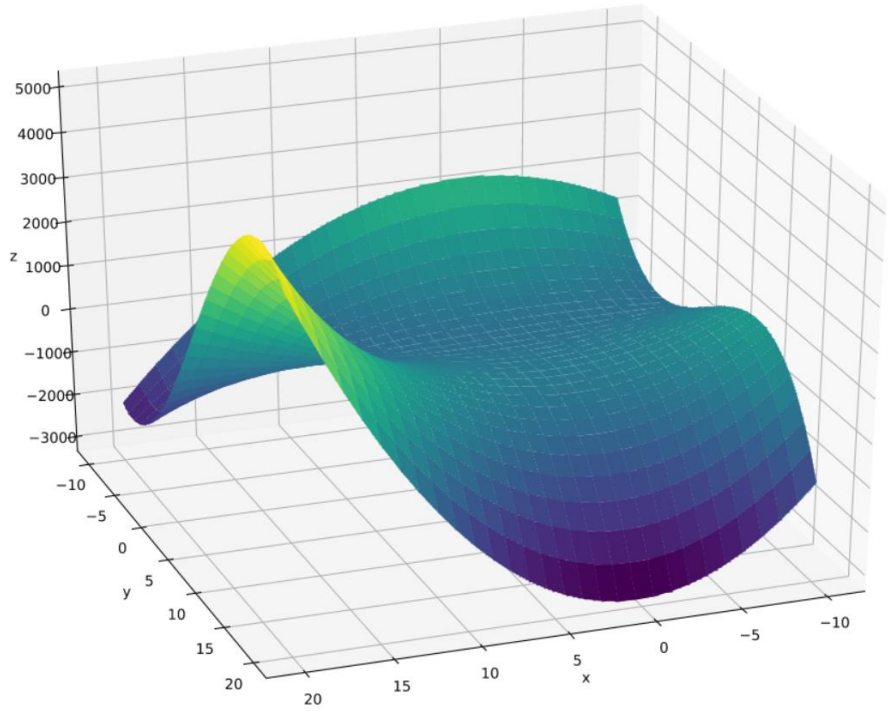
$$f(a,b) = a^2b - b^3 - 11a + 12b^2 + 15$$

1. **Error function** to minimize. In this case f(a,b) as defined above.
2. **Starting point**. We start at **(0,0),** but any other point would work.
3. **Stopping criterion**. The goal is to minimize the function f, which is equivalent to finding a solution for which the gradient is zero. It **is** possible to never reach zero and running into an endless loop. Alternatively, we can stop the algorithm after a fixed number of iterations = **5**. Note: if the error function is known to be strictly positive, in addition the algorithm can be stopped when reaching a predefined error value (possibly a very low one e.g. 10^{-4}).
4. **Learning Rate**. A small learning rate can reduce the chance of jumping over a minimum, but it may be too slow. A large learning rate can produce too large steps and overshoot the local minimum, but it moves faster towards the minimum. We use $\eta = 10^{-2}$. However, adaptive learning rates might be better. In this case the learning rate would be a decreasing function of time.

Explain how you can minimize the following formula using gradient descent:

$$f(a,b) = a^2b - b^3 - 11a + 12b^2 + 15$$

- Let‘s visualize the formula



Note that the global minimum is minus infinite. Therefore we have to manually stop calculations after a fixed number of iterations

Gradient Descent

Explain how you can minimize the following formula using gradient descent:

$$f(a,b) = a^2b - b^3 - 11a + 12b^2 + 15$$

- In every step we move our position in the opposite direction of the gradient (The gradient is always showing towards the steepest ascent, hence we go in the opposite direction in the hope for a minimum).
- Let's calculate the gradient of f(a,b)

$$\nabla f(a,b) = \begin{pmatrix} \frac{\partial f(a,b)}{\partial a} \\ \frac{\partial f(a,b)}{\partial b} \end{pmatrix}$$

$$\frac{\partial f(a,b)}{\partial a} = 2ab - 11$$

$$\frac{\partial f(a,b)}{\partial b} = a^2 - 3b^2 + 24b$$

Explain how you can minimize the following formula using gradient descent:

$$f(a,b) = a^2b - b^3 - 11a + 12b^2 + 15$$

Pseudocode for the developers:

$a \leftarrow 0$
$b \leftarrow 0$
$\eta \leftarrow 10^{-2}$
$\boldsymbol{for}\ i = 1 \ldots nrOfIterations\ do$
  $newA \leftarrow a - \eta * (2ab - 11)$
  $newB \leftarrow b - \eta * (a^2 - 3b^2 + 24b)$
  $a \leftarrow newA$
  $b \leftarrow newB$
$\boldsymbol{end\ for}$

Explain how you can minimize the following formula using gradient descent:

$$f(a,b) = a^2b - b^3 - 11a + 12b^2 + 15$$

- **Step 0.** $a = 0$, b $= 0$, $f(a,b) = 15$, $\eta = 10^{-2}$

- #1
- **Step 1**. $f(a,b) = 15$
- **Step 2**. Gradient in (0,0): $\nabla f(0,0) = \begin{pmatrix} 2ab - 11 \\ a^2 - 3b^2 + 24b \end{pmatrix} = \begin{pmatrix} -11 \\ 0 \end{pmatrix}$
- **Step 3.** New values for (a,b):

$$new \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} - \eta * \nabla f(a,b) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 10^{-2} \begin{pmatrix} -11 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.11 \\ 0 \end{pmatrix}$$

$$\nabla f(a,b) = \begin{pmatrix} 2ab - 11 \\ a^2 - 3b^2 + 24b \end{pmatrix}$$

- #2
- **Step 1**. $f(a,b) = f(0.11, 0) = 13.79$
- **Step 2**. Gradient in (0.11,0): $\nabla f(0.11,0) = \begin{pmatrix} -11 \\ 0.0121 \end{pmatrix}$
- **Step 3.** New values for (a,b):

$$new \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} - \eta * \nabla f(a,b) = \begin{pmatrix} 0.11 \\ 0 \end{pmatrix} - 10^{-2} \begin{pmatrix} -11 \\ 0.0121 \end{pmatrix} = \begin{pmatrix} 0.22 \\ -0.000121 \end{pmatrix}$$

- #3
- **Step 1**. $f(a,b) = f(0.22, -0.000121) = 12.58$
- **Step 2**. Gradient in (a,b): $\nabla f(0.22, -0.000121) = \begin{pmatrix} -11.00005 \\ 0.045496 \end{pmatrix}$
- **Step 3.** New values for (a,b):

$$new \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} - \eta * \nabla f(a,b) = \begin{pmatrix} 0.22 \\ -0.000121 \end{pmatrix} - 10^{-2} \begin{pmatrix} -11.00005 \\ 0.045496 \end{pmatrix} = \begin{pmatrix} 0.33 \\ -0.000576 \end{pmatrix}$$

Gradient Descent

- #4
- **Step 1**. $f(a,b) = f(0.33, -0.000576) = 11.37$
- **Step 2**. Gradient in (0.33,-0000576): $\nabla f(\mathrm{a},\mathrm{b}) = \begin{pmatrix} -11.00038 \\ 0.095075 \end{pmatrix}$
- **Step 3.** New values for (a,b):

$$new \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} - \eta * \nabla f(a,b) = \begin{pmatrix} 0.4400038 \\ -0.001527 \end{pmatrix}$$

- #5
- **Step 1**. $f(a,b) = 10.16$
- **Step 2**. Gradient in (a,b): $\nabla f(\mathrm{a},\mathrm{b}) = \begin{pmatrix} -11.001344 \\ 0.156948 \end{pmatrix}$
- **Step 3.** New values for (a,b):

$$new \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} - \eta * \nabla f(a,b) = \begin{pmatrix} 0.55 \\ -0.003096 \end{pmatrix}$$

Exercise 4
Overfitting

Overfitting

a) *Explain the term overfitting*

b) *Name one example where overfitting can occur*

c) *How can you test if your results suffer from overfitting? (name one idea)*

d) *Name some techniques to avoid overfitting*

Overfitting

a) Explain the term overfitting

- When we say that the model overfits the data, it means that the learned model is fitting the training data too well, including noise and errors, and does not generalize to real-world new data: The model's prediction error on new data from the same distribution will be higher than on the training data.
- This happens when the model has too many free parameters with respect to the complexity of the data and can easily adjust them to all the particularities of the training set.

Overfitting

b) Name one example where overfitting can occur

- I have a dataset with customer data, including the customer unique IDs, and I want the model to predict which customer will like a given product (customer propensity). Sometimes too big models learn to predict on the basis of the customer unique ID.
- This datum will not be available in new data (every customer has a different unique ID). The model here has learned that customer 00102030 likes chocolate, not that any customer with similar characteristics to customer 00102030 might also like chocolate. It has rote-learned the particularities of the training set and cannot generalize to new data.

c) How can you test if your results suffer from overfitting? (name one idea)

- **Idea 1:**
- Split the data into multiple subsets and use only some to learn the model: if performances on different sets are not differing a lot, the model is probably not affected by overfitting

- **Idea 2** (not always applicable):
- Ask an expert to take a look at your classificator

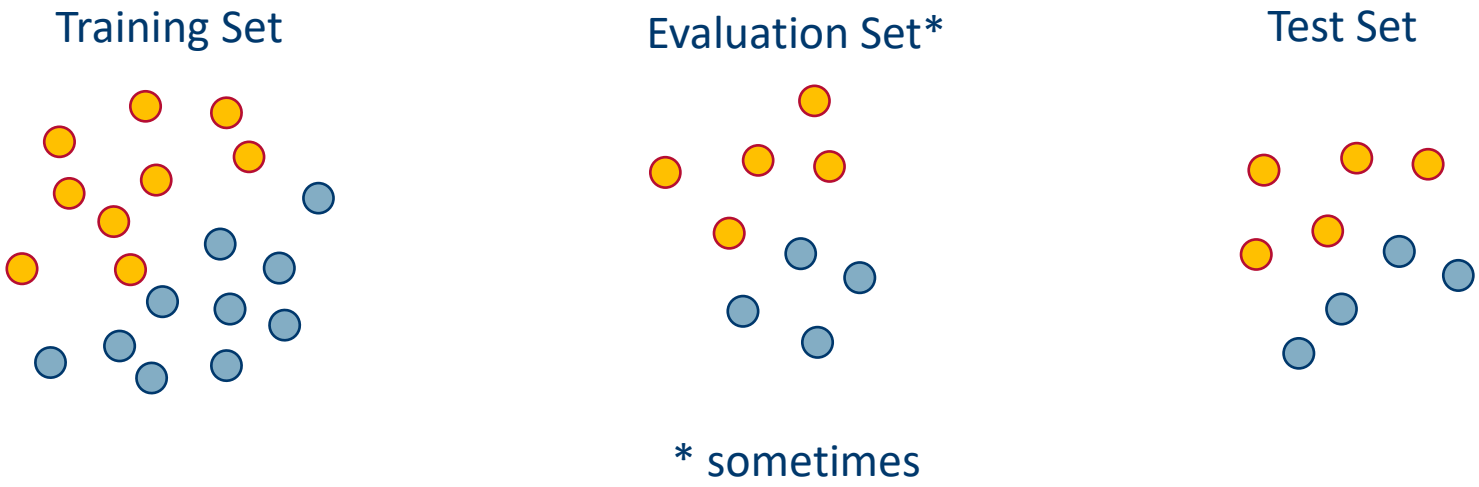
d) Name some techniques to avoid overfitting

- Many techniques to avoid overfitting (basically by keeping the model small):
  - Regularization
  - Pruning
  - Use majority vote on different models trained on similar but not identical training sets
  - Insert randomness in training (dropout layer in deep learning networks)

Exercise 5
Training Set vs. Test set & Cross-validation

Training set vs Test set & Cross-Validation

- *In the lecture we explained that the data is mostly split in two parts (training and test data). Using the whole data maybe builds a better classifier.*
  - *a) Why is it a good idea to split the data? Which negative side effect could be reduced?*
  - *b) Explain what cross-validation is and why it is useful*



Training set vs Test set & Cross-Validation

a) *Why is it a good idea to split the data into train and test set? Which negative side effect could be reduced?*

- Evaluating a model on a set of new data (test set) gives a more realistic measure of the model quality than evaluating it on the training data. Usually, the quality measure on the model is higher on the training set than on the test set.
- Training set and test set should never overlap. If data are shared in between the two datasets, the measure of the model quality is not reliable anymore. This is known as the data leakage problem.

b) Explain what cross-validation is and why it is useful

- Cross-validation creates k subsets of roughly equal size from the original dataset. At each iteration, the model is trained on k-1 subsets and tested on the remaining subset. The test subset is always a different one.
- The error mean gives us a realistic estimate of the model performance on data from this distribution. The error variance gives us a measure of the homogeneity of the dataset.

Exercise 6
Building a Model

Building a model

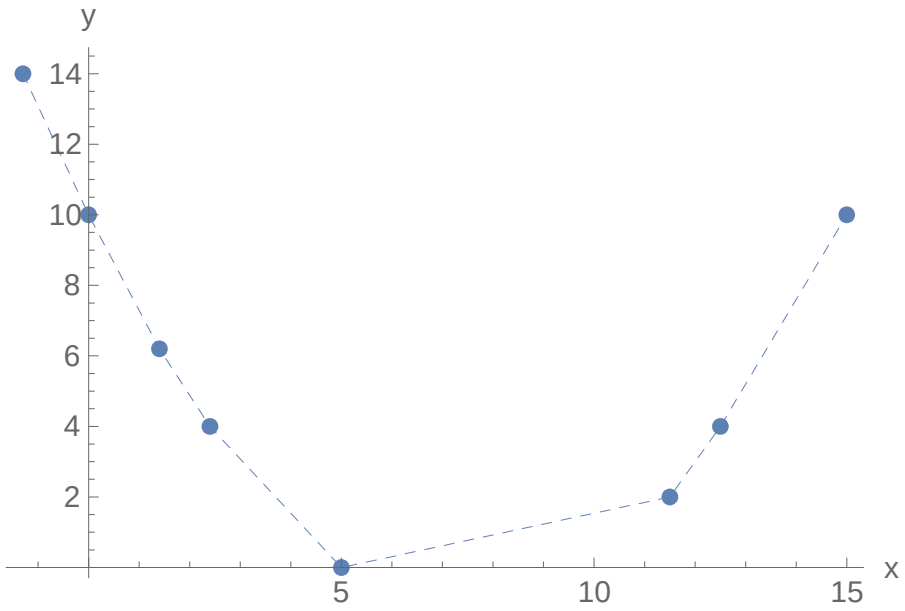
- *In this exercise we will go through the whole process of building a model.*
- *Take a look at the following x-y data set.*

x	-1.3	0.0	1.4	2.4	5.0	11.5	12.5	15.0
y	14.0	10.0	6.2	4.0	0.0	2.0	4.0	10.0

- *The goal is to train a model that can predict y values for given x values.*
- *Please consider the three following requirements*
  - ***Model class***
  - ***Score function***
  - ***Algorithm for Model Fitting***

- ***Model class**: Our model can be described by a parable which maps the x value to the y value. The parable formula can be generalized to*

$$y = 0.2\,(x - a)(x - b)$$



- ***Score function**: Define the mean squared error E for the data set, with respect to the chosen model.*

- ***Algorithm for Model Fitting**: Let's use the gradient method*
  a) *What is the most important assumption for the error function, to be able to apply the gradient descent method?*
  b) *Explain the most important steps for the gradient descent methods.*
  c) *Calculate the gradients used for our learning goal.*
  d) *Write down the update functions for our learning goal.*
  e) *Explain two possible problems with the usage of a constant step width in gradient descent. Give one example to resolve the step width problem.*

***Score function**:*
Define the mean squared error E for the data set, with respect to the chosen model.

$$MSE = \frac{1}{8}\sum_{i=1}^{8}(y_i - f(x_i))^2 = \frac{1}{8}\sum_{i=1}^{8}(y_i - 0.2(x_i - a)(x_i - b))^2$$

***Algorithm for Model Fitting**: Gradient method*

a) What is the most important assumption for the error function, to be able to apply the gradient descent method?

The error function has to be differentiable in the parameters a and b.

***Algorithm for Model Fitting**: Gradient method*

b) Explain the most important steps for the gradient descent methods.

– Choose a starting point $(a_{i=0}, b_{i=0})$ and a learning rate η

1. Calculate the error function in the current parameter point.
2. Calculate the value of the partial derivatives for all parameters in current parameter point
3. Update the parameters as $(a_{i+1}, b_{i+1}) = (a_i, b_i) - \eta \nabla E$
4. Repeat from 1. for a number N of iterations or till error below threshold or no change observed anymore in parameter update

***Algorithm for Model Fitting**: Gradient method*

c) Calculate the gradients used for our learning goal

$$\frac{\partial MSE}{\partial a} = \frac{2}{8}\sum_{i=1}^{8}\big(f(x_i) - y_i\big)0.2(-x_i + b)\big)$$

$$\frac{\partial MSE}{\partial b} = \frac{2}{8}\sum_{i=1}^{8}\big(f(x_i) - y_i\big)0.2(-x_i + a)\big)$$

***Algorithm for Model Fitting**: Gradient method*

d) Write down the update functions for our learning goal.

$$a_{i+1} = a_i - \eta \frac{\partial MSE}{\partial a}(a_i, b_i) = a_i - \eta \frac{2}{8} \sum_{k=1}^{8} \big(f(x_k) - y_k\big) 0.2(-x_k + b_i)\big)$$

$$b_{i+1} = b_i - \eta \frac{\partial MSE}{\partial b}(a_i, b_i) = b_i - \eta \frac{2}{8} \sum_{k=1}^{8} \big(f(x_k) - y_k\big) 0.2(-x_k + a_i)\big)$$

***Algorithm for Model Fitting**: Gradient method*

e) Explain two possible problems with the usage of a constant step width in gradient descent. Give one example to resolve the step width problem.

- If the step width is chosen too big, the minimum can be skipped. If it's chosen too small the minimum can never be reached.
- Method to resolve: Use an adaptive step width as a decreasing function of time.

Exercise 7
Search Strategies

Search Strategies

Describe the following optimization strategies for a function y=f(x)

- *Grid search*
- *Random search*
- *Hill climbing*
- *Bayesian optimization*

Search Strategies

Describe the following optimization strategies for a function y=f(x)

- **Grid Search** is a brute-force strategy. All points in the domain of x are evaluated. The optimum value of f(x) is then selected. Limited to few dimensions
- **Random Search**: Only N random points in the domain of x are evaluated. The optimum value of f(x) is then selected. Precision of optimum depends on value of N. Faster than grid search (esp. on many parameters).

Search Strategies

Describe the following optimization strategies for a function y=f(x)

- **Hill Climbing**. This is a greedy strategy. We start from a point at random and instead of evaluating many others, we evaluate only points in the neighborhood. If no points in the neighborhood do better, we stop. Of course faster, but it can get stuck in local optima and miss the absolute optimum.
- **Bayesian Optimization**. Consists of 2 phases.
  - Phase 1 (warm up): Select N random points for x
  - Phase 2: Build a surrogate model on these N random points, like $P(output|\ points)$ and find optimum for surrogate model. Using one of previous methods

  Faster (surrogate model is usually simpler). Precision of evaluation depends on quality of surrogate model.

  Can use active learning to improve the model.

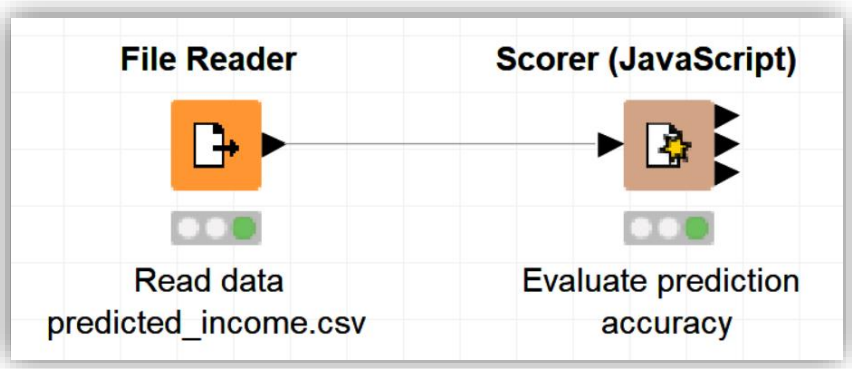
Exercise 8
Practice with KNIME

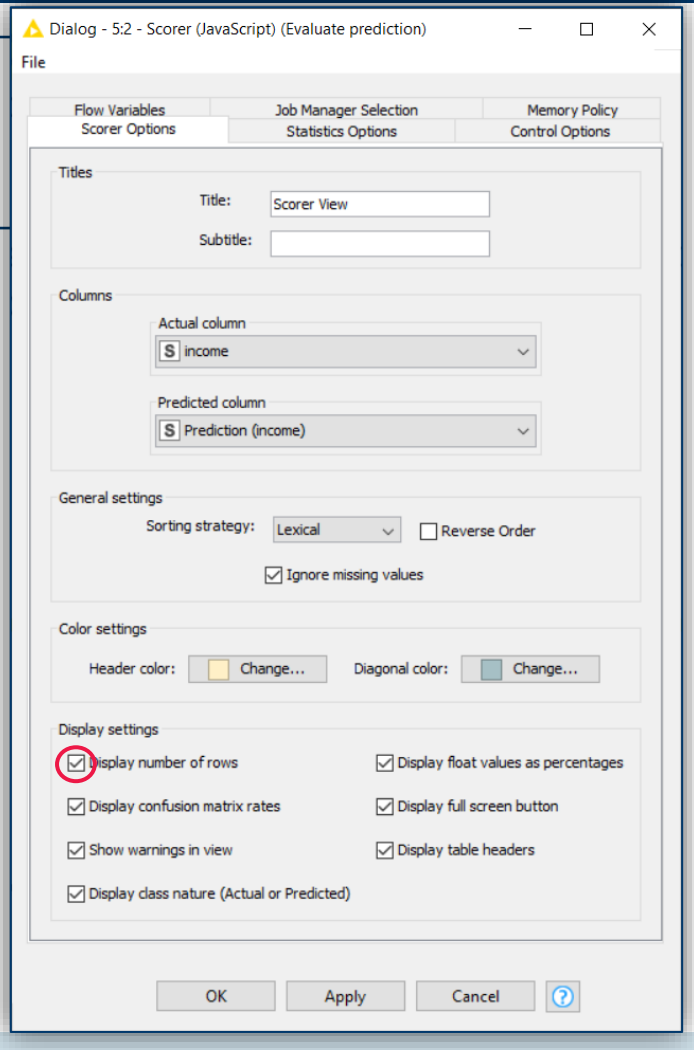
1. Confusion Matrix

1. Read data ***predicted_income.csv***
    1. The "income" column contains people's actual income class values. The "Prediction (income)" column contains their predicted income class values produced by some classification model based on the other information available in the dataset. The income class has two values: "<=50K" and ">50K"
2. Evaluate the accuracy of the income class prediction using the Scorer (JavaScript) node.
3. Execute the node, and open the interactive view.
    a) How many events are correctly classified in total?
    b) How many events are wrongly classified in total?
    c) Out of how many data rows in the dataset?
    d) What is the overall accuracy of the model?
    e) And Cohen's kappa?
4. Open the configuration dialog of the Scorer (JavaScript) node and enable the displaying of the class statistics table.

1. Confusion Matrix

Evaluate the accuracy of the income class prediction using the Scorer (JavaScript) node.





1. Confusion Matrix

Scorer View

Confusion Matrix

Rows Number : 6513	<=50K (Predicted)	>50K (Predicted)	
<=50K (Actual)	4403	542	89.04%
>50K (Actual)	521	1047	66.77%
	89.42%	65.89%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
83.68%	16.32%	0.556	5450	1063

a) How many events are correctly classified in total? **4403 + 1047 = 5450**

b) How many events are wrongly classified in total? **521 + 542 = 1063**

c) Out of how many data rows in the dataset? **6513**

d) What is the overall accuracy of the model? **83.68%**

e) And Cohen's kappa? **0.556**

2. ROC Curve

1. Read data ***predicted_gender.csv***

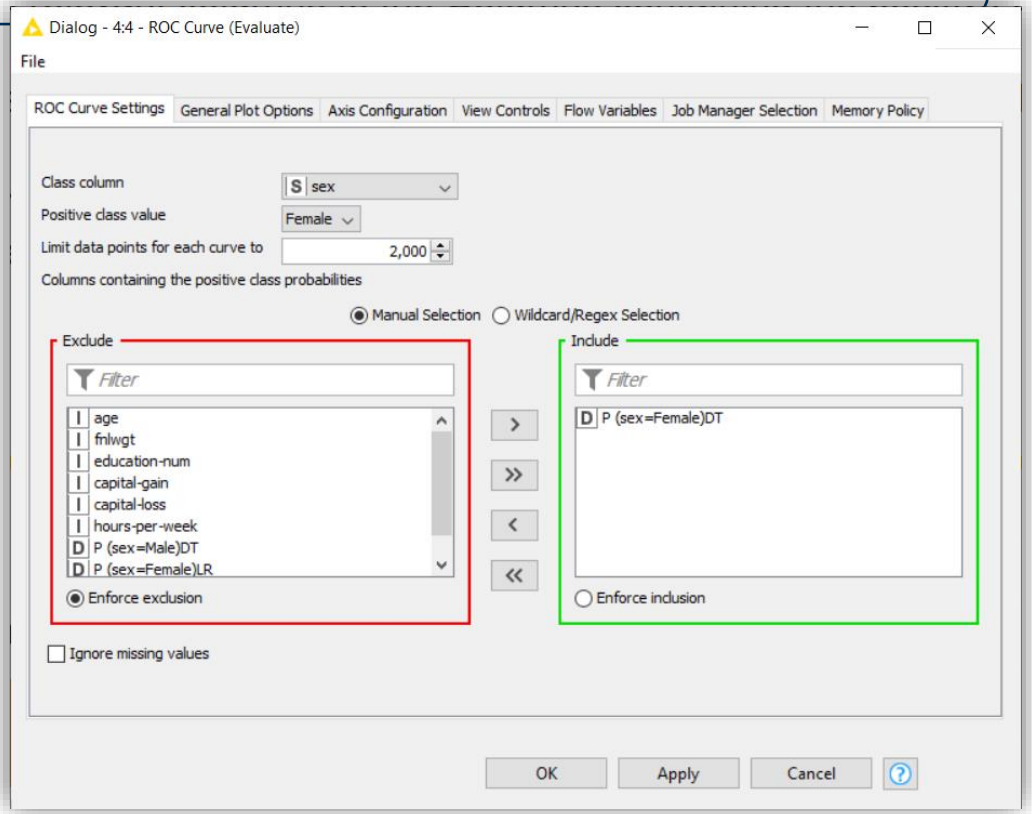
   The "sex" column contains people's actual gender: Female or Male. The "Prediction (sex) ..." columns contain their gender values predicted by two different classification models - a decision tree (DT) and logistic regression model (LR). The "P(sex=Female)..." columns contain the predicted probabilities of being female produced by the two models.

2. Evaluate the performance of the decision tree model using ROC curve node
   1. Set Class column, Positive class value, and Columns containing the positive class probabilities in the configuration dialog
   2. Execute the node and open the interactive view
   3. What is the area under the curve for the decision tree model?

3. Compare the performance of the decision tree and logistic regression models by plotting their ROC curves in the same graph
   1. Open the configuration dialog of the ROC Curve node
   2. Add the relevant columns to the Columns containing the positive class probabilities
   3. Which of the models perform better?
   4. What is the area under the curve for the logistic regression model?

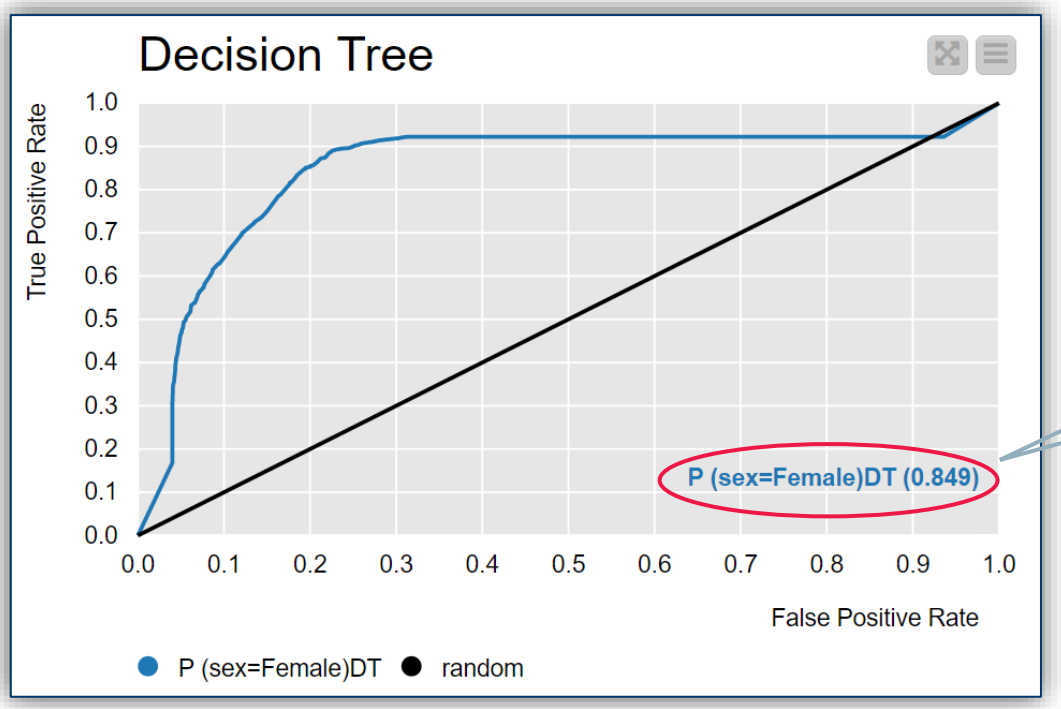
2. ROC Curve

- Evaluate the performance of the decision tree model using ROC curve node



2. ROC Curve

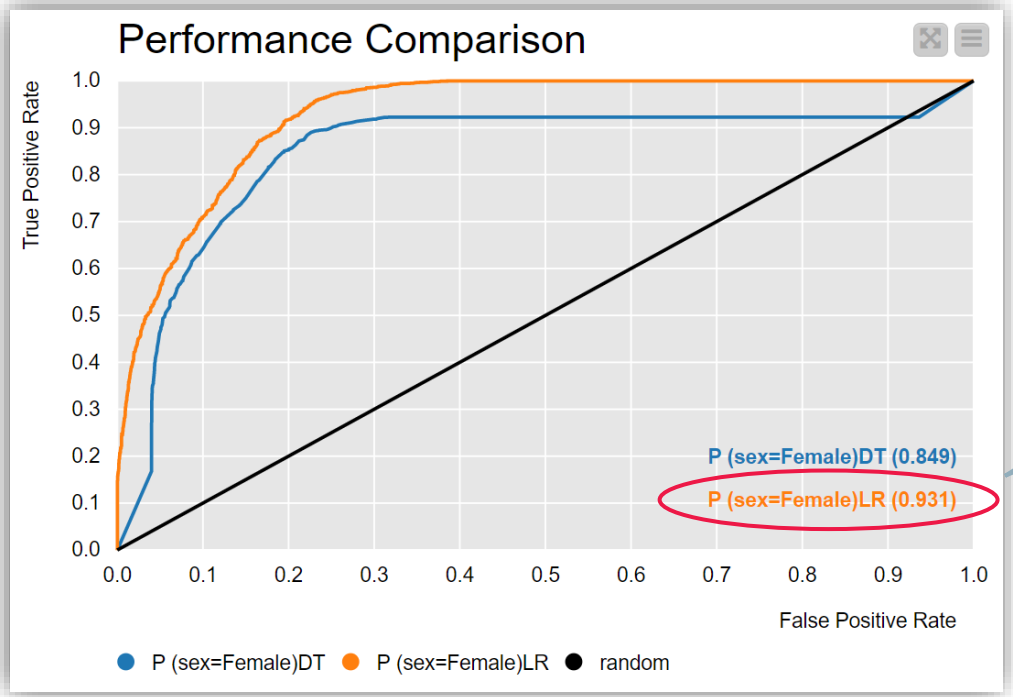
– What is the area under the curve for the decision tree model?



AUC Decision Tree: 0.849

2. ROC Curve

- Compare the performance of the decision tree and logistic regression models by plotting their ROC curves in the same graph



AUC Logistic Regression: 0.931

The logistic regression model performed better

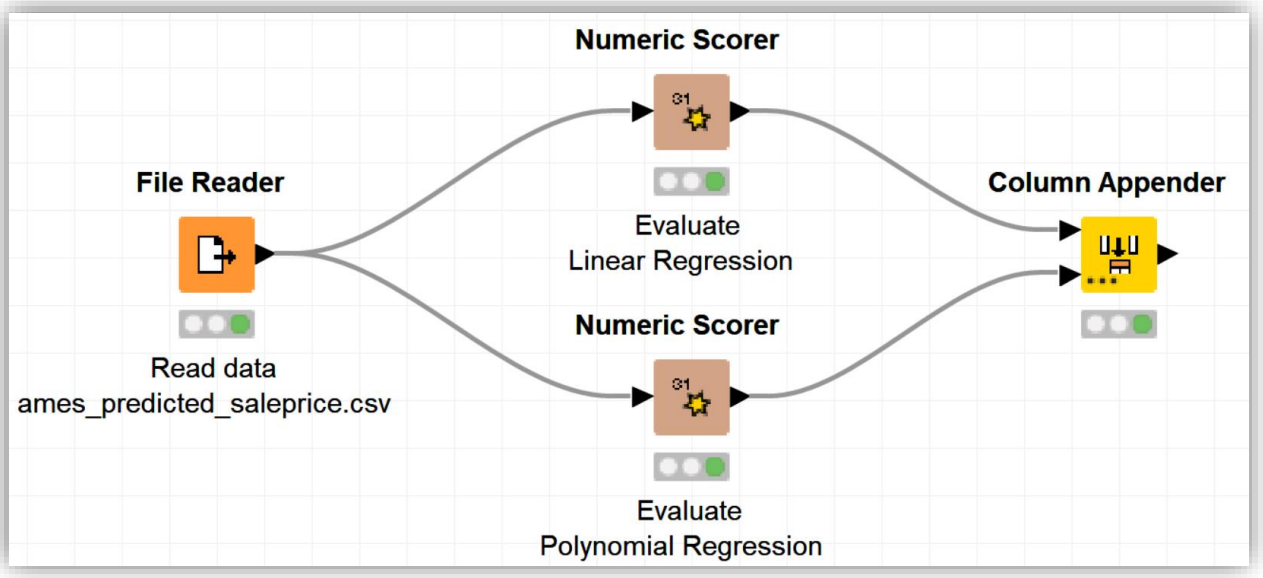
3. Numeric Scorer

1. Read the dataset ***ames_predicted_saleprice.csv***. It contains information about houses in Ames (Iowa, USA).
   - The "SalePrice" column contains the actual sale price.
   - The “Prediction (Saleprice)LR” column contains the price predicted by a Linear Regression model.
   - The “Prediction (Saleprice)PR” column contains the price predicted by a Polynomial Regression model.

2. Evaluate the accuracy of the two models. Use two Numeric Scorer nodes. Group the error measures in a single table using the Column Appender node
   - Which Error Measures are available?
   - Which model performed better and why?

3. Numeric Scorer

Evaluate the accuracy of the two models. Use two Numeric Scorer nodes. Group the error measures in a single table using the Column Appender node



- Which Error Measures are available?
- Which model performed better and why?

Row ID	D Prediction (SalePrice)LR	D Prediction (SalePrice)PR
R^2	0.61	0.708
mean absolute error	30,687.181	28,477.098
mean squared error	2,289,544,473.136	1,712,419,345.865
root mean squared error	47,849.185	41,381.389
mean signed difference	1,769.815	824.005
mean absolute percentage error	0.191	0.178

The PR model has better performance according to each measure

Thank you