Handout on basic machine learning

The following exercises should help fix some topics presented during the lessons, without being in front of a computer and possibly together with some colleague and some coffee. With one or more coffee symbols are indicated the exercises that may require some more time/thought.

1 Classification metrics

1. Precision and recall in binary classification.
In binary classification it is custom to distinguish a positive (1) and negative (0) classes. Precision and recall metrics are often defined in this setting with respect to the positive class, especially when this is the minority class. You are given the following confusion matrix:

$$C = \begin{pmatrix} 26 & 16 \\ 33 & 25 \end{pmatrix}$$

Compute in this binary setting the precision and recall, where the rows/columns report results for the two classes [0,1], in this order. Try not to use the formulas given in the class, but to remember the meaning of precision and recall and 'rederive' those formulas.

2. Aggregated metrics in multiclass classification (☕).
In a multiclass framework you are given the following confusion matrix:

$$C = \begin{pmatrix} 11 & 11 & 9 \\ 9 & 9 & 11 \\ 14 & 10 & 16 \end{pmatrix}$$

where the rows/columns report results for the three classes [0,1,2], in this order.

- Compute precision and recall for each class. Report the mean of precision and recall over all classes. This way of combining individual metrics to return a final one is called "macro"-averaging.
- Evaluate also the overall precision of the model as the total percentage of correct predictions. This way of aggregating metrics is called "micro"-averaging.

BONUS: you may want to use numpy manipulations to perform explicitely the task. Can you find in sklearn what are functions doing the heavy work for you? Can you check your results using sklearn function?

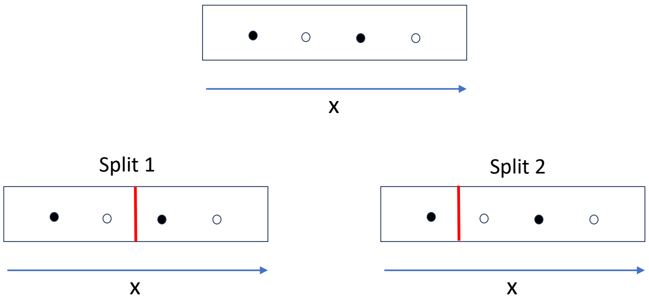

Figure 1: Figure for Ex. 5

2 Regression and classification models

3. Linear regression.
Given a single input continuous feature x and an output y, select which models can be fit via linear regression:

1. $y = ax + b$
2. $y = ax$
3. $y = ax^2 + bx$
4. $y = ax^b + cx$
5. $y = ae^{-x} + b$
6. $y = ae^{-bx} + c$
7. $y = ae^{-bx} + ce^{-dx}$

4. Linear regression continued.
Suppose you still want to use a modified scheme, still based on linear regression, to fit the functions that you selected as problematic in the previous exercise. How could you do that?

5. Splits in classification trees (☕).
You are given a training data set of 4 points as in Fig. 1 where black points belong to one class and white to another. Note that you have just a single scalar feature. Between the two suggested splits can you indicate which one could lead to a positive information gain? Choose the entropy or Gini criterion and evaluate numerically (at least in formulas) the information gain of the most promising split.

6. Splits in regression trees (☕).
You are given a training data set as in Fig. 2 with a single continuous input feature x.

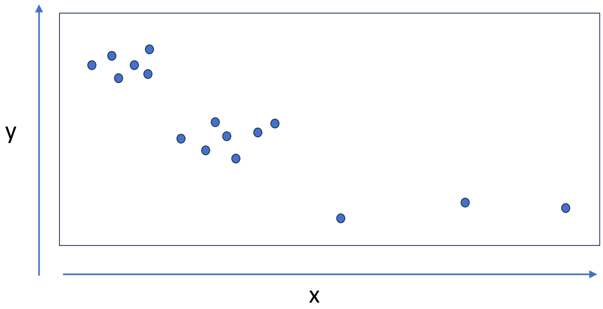

Figure 2: Figure for Ex. 6

1. Suppose you have a tree without any split. What would be the tree prediction if you want to minimize the SSR (sum of squares of the residuals)? HINT: a tree without splits is associated to a constant prediction c. Show that the optimal c is given by the mean of the y_i by analytically minimizing the SSR. Note that you have to perform some derivatives here. This justifies the choice, presented during the lesson, of the mean as best value for each leaf of a regression tree.

2. Now suppose you build a tree of depth 1 (called also a *decision stump*). Draw in Fig. 2 a generic decision boundary for the decision stump and the associated prediction values in the two splitting regions. Can you draw approximately the location of the optimal decision boundary? HINT: the splitting criterion for regression chooses the boundary such that the resulting SSR is as small as possible.

3. Now suppose to find an optimal tree of depth 2. How would the decision boundaries look like?

7. Gini criterion (☕).

Motivate the Gini criterion. In particular consider one set S of N points with associated classes $c_i, i = 1..N$. Take the sample proportions p_c for each class as defined in the lesson. Now propose the following labelling scheme that uses just the sample proportions:

1. choose a point at random from S

2. assign that point to class c with probability p_c

Show that the probability of labelling in a wrong way the the point with this procedure is exactly $G(S)$. The Gini index can be therefore interpreted as giving a measure of how well the sample proportions describe the points inside the corresponding leaf.

8. Bias-Variance decomposition (☕ ☕).

Suppose that you want to fit a deterministic mapping $f^{ex}(x), x \in X, f^{ex}$:

$X \to \mathbf{R}$. A parametric fitted model will provide some predictions $Y^{model}(x) \in \mathbf{R}, x \in X$. These can be considered stochastic (random), because fitting with a different training set could have provided (hopefully slightly) different results. As discussed in the class, this is called the *variance* of the model.
The error at x can be defined as:

$$Err(x) \equiv E[(Y^{model}(x) - f^{ex}(x))^2] \tag{1}$$

, where the symbol $E[]$ is used to indicate the expectation value. We also define the function $h(x) \equiv E[Y^{model}(x)]$.

- Prove that:

  $$Err(x) = E[(Y^{model}(x) - h(x))^2] + (h(x) - f^{ex}(x))^2 \tag{2}$$

  , where the first term is the variance contribution to the error and the second is the bias.

- Interpret the archery figure presented in the lessons according to the formula above.

9. Change of data representation (☕☕).
Point 1. Consider a fixed training set $(\boldsymbol{X}_1, y_1), .., (\boldsymbol{X}_N, y_N)$. A normalization or standardization amounts to the replacement:

$$X_{i,a} \to \tilde{X}_{i,a} = m_a X_{i,a} + n_a, \tag{3}$$

where the index a indicates the corresponding feature and m, n are constants. Prove that fitting a linear model using the standardized/normalized variables will provide the same predictions on the test set. Hint: show that you can obtain the same predictions from the two models after changing the definition of the parameters. Do you expect the same to hold with regularization?
Point 2. Consider instead now the more generic transformation:

$$X_{i,a} \to \tilde{X}_{i,a} = g_a(X_{i,a}), \tag{4}$$

where the $g_a(x)$ are monotonic real functions. Show that two decision stamps (trees with depth 1) based on the same splitting criterion will provide equivalent models after the data transformation. Argue that the same conclusion is true for any decision tree and a random forest model. Hint: the criterion for choosing a cutoff to split a node along feature a is based only on the values of y_i and the ordering of $\{X_{i,a}\}_{i=1..N}$.

10. Linear regression with categorical variables (☕☕☕).
When an input feature is categorical and with a natural ordering, like {bad, good, super} we have two possibilities to use the variable as input of a linear model. We want to get some intuition considering the case of a single input categorical variables x with n-levels, and a scalar output y.
Point 1. One possibility is to map the categorical input to a scalar one, e.g. {bad → 0}, {good → 1} and {super → 2}.

- Can you write down how the most generic linear model would look like in this setting? In particular how many parameters would the model have?

Point 2. Another possibility is to use a one-hot encoding. This would encode the categorical variable x into a one-hot vector $(x_1, ..., x_n)$ with length equal to the number of categories and $x_i \in \{0, 1\}$. This vector would define n variables out of the single categorical one.

- Show that $x_1, ..x_n$ are not independent. As a consequence the minimization problem solved by linear regression is not well posed. One trick to circumvent this issue is to drop the last variable from the input features.

- Can you write down how the most generic linear model would look like in this setting? In particular how many parameters would the model have?

- **HARDER**: Do you know in advance what would be the result of performing linear regression in this setting? Hint: the model should predict for each category the mean $\frac{1}{N_c}\sum_{i|x_i=c} y_i$, that is the mean over all predictions for that category. Try to write down the residual some of squares for this model and differentiate w.r.t. to the parameters you found in the previous question.

BONUS: inspect what happens when you combine a categorical feature variable with a continuous one, especially in the one-hot encoding setting? How would you interpret the coefficients?

3 Train-test split

11. Possible mistakes using wrong test-set distributions (☕☕☕).

Sometimes to avoid information leakage one is forced to use a test set with a different distribution than the training set. This exercise builds some intuitions on how metrics like precision and recall will be affected.

Consider a binary classification framework. As a setup we consider a starting population X, with X_p and X_n the subsets of positive and negative examples. We consider a fixed classifier C (also called *discriminator*) assigning to each eelement of the population a label and define $P(C = p|E = p)$ as the conditional probability that if an element is positive, the classifier will be also positive. Similarly one can define $P(C = n|E = p)$,$P(C = p|E = n)$ and $P(C = n|E = n)$.

Let a test set population be built with the following process:

1. Choose a random number R with a $Ber(\alpha)$ distribution, i.e. 0 with probability α and 1 otherwise.

2. If $R = 0$ choose an element from X_n (according to the corresponding conditional distribution). If $R = 1$ choose an element from X_p.

Let the resulting modified, unbalanced, population be denoted by X_α.

Answer the following questions:

- In the test set population X_α, what is the probability for an element to be negative/positive?

- Write for large N a confusion matrix for the test set X_α in terms of the four conditional probabilities.

- Use the previous results to derive precision and recall as a function of α.