Machine Learning Exercise 1

Alper Hamza Ari-5771973, Chenqi Hao-5781375,
Danil Skokov-5779466, Said Orfan Haidari-5781295

October 2023

1 General Questions

1.1 What are the stages of the ML cycle? Which ones are iterative stages?

1. Preprocessing: Collecting data, cleaning it from noise, and formatting it correctly, etc.

2. Feature Extraction and Encoding: Selecting relevant features from preprocessed data or transforming the features to create informative data that our machine learning model can use. One-hot encoding is useful for categorical data.

3. Feature Selection: Selecting the subset of features that are relevant for our purpose.

4. Machine Learning: Training the model with the final form of the dataset created during the previous stages.

5. Evaluation and Model Selection: Testing and evaluating the model using various methods like cross-validation or a separate validation dataset. Adjustments may be needed based on the results.

6. Post-Processing: Adjusting the model's results for fairness or formatting the output as needed.

Since creating a good machine learning model often requires iterative steps like preprocessing, adjusting data, feature extraction, model training, and algorithm selection, all stages of machine learning design can be considered iterative and form a cycle.

1.2 What are the different types of learning?

Learning can be categorized into:

- Supervised learning : Algorithm learns from a labeled dataset (Ground Truth). e.g. classification.
- Unsupervised learning : Algorithm is trained on a dataset that contains only input data without the corresponding outputs.

There is also a learning type that combines elements of both, known as Semi-Supervised learning.

1.3 How would you describe the overfitting and underfitting phenomenon?

Overfitting occurs when a model is overly complex and fits not only the data but also the noise in the data. The model is too rigid for the data and it can't generalize it. Underfitting occurs when the model is too simple to capture the complexity of the data.

2 Naive bayes

2.1 Probabilities

$$\begin{aligned} P(\text{yes}) &= \frac{6}{10} \\ P(\text{red}|\text{yes}) &= \frac{3}{6} \\ P(\text{grand tourer}|\text{yes}) &= \frac{2}{6} \\ P(\text{domestic}|\text{yes}) &= \frac{2}{6} \\ P(\text{no}) &= \frac{4}{10} \\ P(\text{red}|\text{no}) &= \frac{1}{4} \\ P(\text{grand tourer}|\text{no}) &= \frac{2}{4} \\ P(\text{domestic}|\text{no}) &= \frac{3}{4} \end{aligned}$$

2.2 Predict the probability that a car with properties x1 = red,x2 = grand tourer, x3 = domestic will be stolen.

$$P(\text{yes} \mid \text{red, grand tourer, domestic}) = \frac{P(\text{yes}) \cdot P(\text{red, grand tourer, domestic} \mid \text{yes})}{P(\text{red, grand tourer, domestic})}$$

Let's first calculate the numerator since it is the most important part of the equation. The denominator is just for normalization, which we will do later. Since conditional independence is assumed:

$$\begin{aligned} P(\text{yes} \mid \text{red, grand tourer, domestic}) &= P(\text{yes}) \cdot P(\text{red} \mid \text{yes}) \cdot P(\text{grand tourer} \mid \text{yes}) \cdot P(\text{domestic} \mid \text{yes}) \\ &= \frac{6}{10} \cdot \frac{3}{6} \cdot \frac{2}{6} \cdot \frac{2}{6} \\ &= \frac{1}{30} \end{aligned}$$

Now, in order to normalize the result, the normalization coefficient must be calculated.

$$\begin{aligned} P(\text{no} \mid \text{red, grand tourer, domestic}) &= P(\text{no}) \cdot P(\text{red} \mid \text{no}) \cdot P(\text{grand tourer} \mid \text{no}) \cdot P(\text{domestic} \mid \text{no}) \\ &= \frac{4}{10} \cdot \frac{1}{4} \cdot \frac{2}{4} \cdot \frac{3}{4} \\ &= \frac{3}{80} \\ Z &= \frac{1}{30} + \frac{3}{80} \end{aligned}$$

$$P(\text{yes} \mid \text{red, grand tourer, domestic}) = \frac{\frac{1}{30}}{\frac{1}{30} + \frac{3}{80}} = 0.47$$

2.3 In general: What are the benefits, what are the downsides of using Naive bayes?

Advantages:

- Naive Bayes is simple, easy to understand and implement and fast
- It can perform well with small training datasets. This can be beneficial when you have limited data.

Disadvantages:

- Naive Bayes assumes that all features are independent, which is rarely true in real-world data
- When a particular feature-label combination has not been observed in the training data, Naive Bayes assigns it a zero probability

2.4 Derive Equation 1 using Bayes' theorem, the chain rule of probabilities and the conditional independence assumption stated above.

So the Bayes' theorem formula is:

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(y) \cdot P(x_1, x_2, \ldots, x_n|y)}{P(x_1, x_2, \ldots, x_n)}$$

Using the joint probability model and the chain rule of probability, the numerator can be written as:

$$P(y) \cdot P(x_1, x_2, \ldots, x_n|y) = P(y, x_1, x_2, \ldots, x_n) = P(y|x_1, x_2, \ldots, x_n) \cdot P(x_1, x_2, \ldots, x_n)$$

$$= P(y|x_1, x_2, \ldots, x_n) \cdot P(x_1|x_2, x_3, \ldots, x_n) \cdot P(x_2, x_3, \ldots, x_n)$$

After writing every term in this form, if we assume conditional independence, then the numerator becomes:

$$P(y) \cdot P(x_1|y) \cdot P(x_2|y) \cdot \ldots \cdot P(x_n|y)$$

Which makes the new formula:

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(y) \cdot P(x_1|y) \cdot P(x_2|y) \cdot \ldots \cdot P(x_n|y)}{P(x_1, x_2, \ldots, x_n)}$$

Since the denominator is independent of the value of y, it is only used for normalization.

So,

$$Z = P(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{K} P(y = k) \cdot \prod_{i=1}^{n} P(x_i|y = k)$$

The final formula is:

$$P(y|x_1, x_2, \ldots, x_n) = \frac{1}{Z} \cdot P(y = k) \cdot P(x_1|y) \cdot P(x_2|y) \cdot \ldots \cdot P(x_n|y)$$

3 Ranking Losses

3.1 Formulate mathematically a loss function that evaluates how well some generic prediction ˆy matches the target ranking y.

Lets create a binary function that compares the ranking of two items between ground truth and the predicted ranking. If the ranking between these two items are same as the real ranking then the function produces zero and if they are different the function produces one.

$$RankDiff(x_i, x_j) = \begin{cases} 0 & \text{if real rankings and predicted rankings are the same} \\ 1 & \text{if real rankings and predicted rankings are different} \end{cases}$$

Now for the loss function all the results will be added to each other. The model with the least sum is the better one.

$$Loss = \sum_{i=1}^{N} \sum_{j=i+1}^{N} RankDiff(x_i, x_j)$$

Let's use this function to compare both models for x_1, x_2, and x_3. Firstly, for $\hat{y_1}$, x_1 and x_2 are compared. The real rankings between x_1 and x_2 are 1 and 2, and the predicted rankings are 1 and 3. Since the rankings are preserved, the RankDiff(x_1, x_2) returns 0. The other calculations are shown below.

$$\begin{aligned} RankDiff(x_1,\, x_2) &= 0 && \text{rank preserved} \\ RankDiff(x_1,\, x_3) &= 0 && \text{rank preserved} \\ RankDiff(x_2,\, x_3) &= 1 && \text{rank not preserved} \end{aligned}$$

$$Loss = 0 + 0 + 1 = 1$$

Now we will do the same for $\hat{y_2}$:

$$\begin{aligned} RankDiff(x_1,\, x_2) &= 0 && \text{rank preserved} \\ RankDiff(x_1,\, x_3) &= 0 && \text{rank preserved} \\ RankDiff(x_2,\, x_3) &= 0 && \text{rank preserved} \end{aligned}$$

$$Loss = 0 + 0 + 0 = 0$$

3.2 According to this mathematical formulation, which model is better at ranking? Why is the squared error problematic in this case?

Since the model with the lowest loss is considered to be the better one, model $\hat{y_2}$ is better.

In ranking problems, the true rankings are typically integer values that represent the order of items. The squared differences are not meaningful in this context