# Exercise 3

*Pattern Recognition, Fall 2021*

*Nico Aebischer*
*Max Jappert*

## 3. Theory/Coding questions in general

- The posterior $P(c, x)$ for **Naive Bayes** is calculated by multiplying the prior $P(c)$ for a given class $c$ with the likelihood $P(x, c)$ of the the element $x$ belonging to a given class. The prior $P(c)$ for an element belonging to class $c$ is calculated by dividing the amount of elements in c by the total amount of elements in all classes combined. The likelihood $P(x, c)$ of element x being in class c is computed as the number of times element $x$ appears in class $c$ divided by the total amount of elements in class $c$.

  The posterior $P(y = 1 \mid x)$ for **Logistic Regression** is decided by squashing element $x$ into a value in the range $[0, 1]$. For $x \in \mathbb{R}$ this can be done with the sigmoid function $P(y = 1 \mid x) = \frac{1}{1+e^{-x}}$, which for $x' \in \mathbb{R}^k$ needs to be modified in order to become $P(y = 1 \mid x, w, w_0) = \frac{1}{1+e^{-(w^T x+w_0)}}$. The element $x$ is classified to class $y = 1$ if $P(y = 1 \mid x) > 0.5$, due to the fact that, trivially, $P(y = 0 \mid x) = 1 - P(y = 1 \mid x)$. The same holds for $x' \in P(y = 1 \mid x, w, w_0)$.

- We would probably use the **Naive Bayes** approach, to tackle multi-class classification problems. The alortithm is well known for also performing well in multi class prediction. We can simply compute the probability of each class label in the usual way and then just pick the class with the largest probability. Also, we already know the distribution of the classes. This enables us, to already plot the distributions.
  **Logistic Regression** is limited to two-class problems by default. If one wanted to nevertheless use it for multi-class problems, some modification would be necessary. One such approach would be to split the multi-class classification problem into multiple binary problems. Then, one could evaluate the logistic regression on each of those subproblems. This of course accounts for much more effort that has to be done compared to the NB approach.

## 4. Theory/Coding questions Naive Bayes

- The first assumption for Naive Bayes is that all features of an element to be classified are independent. E.g. the likelihood of a document $x$ to be classified to class $c$ is the likelihood that all words $x_1, ..., x_n$ in document $x$ are in $c$. This is calculated by multiplying the individual probabilities of $x_i$ belonging to $c$. This assumes the following equation:

  $$P(x_1, ..., x_n \mid c) = P(x_1 \mid c) \cdot ... \cdot P(x_n \mid c)$$

  Due to the definition of independence, i.e. that $P(A, B) = P(A)P(B)$ iff $A, B$ are independent, the above formula only holds if $x_1, ..., x_n$ are independent.

  The second assumption is that all features are considered to be equally important to the classification. They aren't weighted.

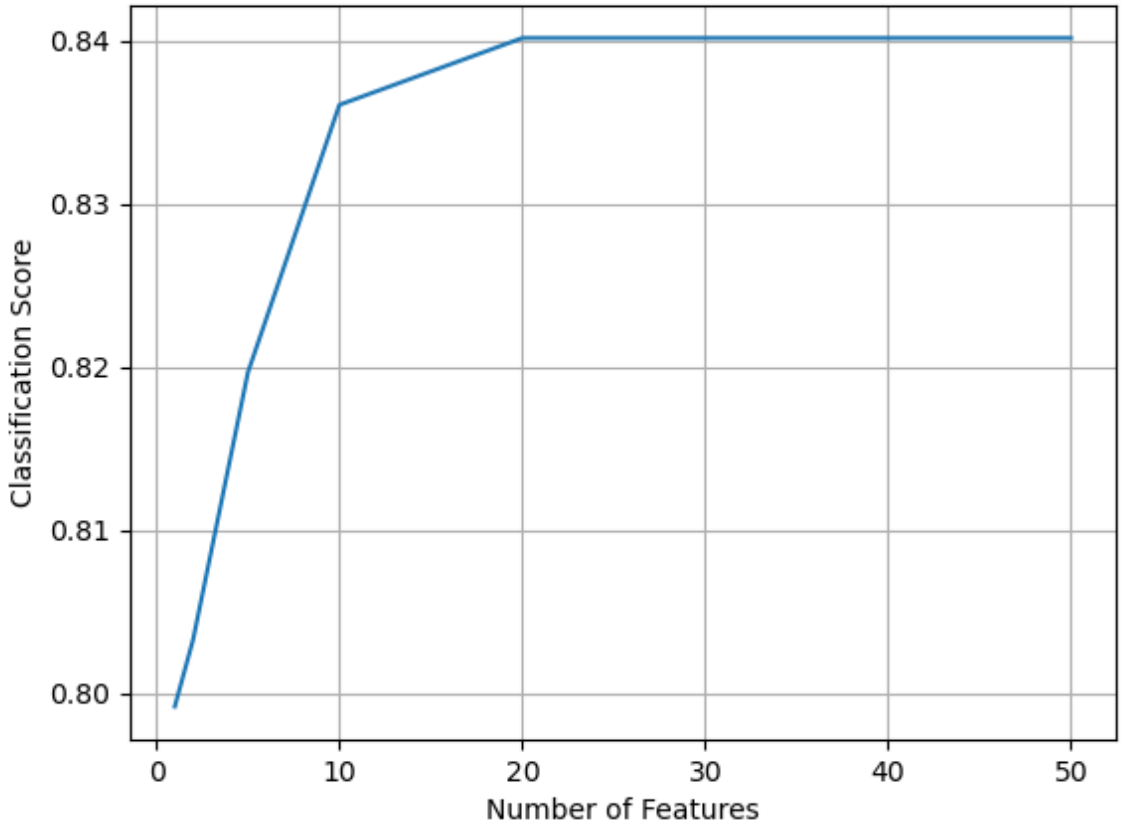
- `3 most popular HAM words:`
  `the`
  `of`
  `to`

```
3 most popular SPAM words:
the
to
you

3 most distinct HAM words:
linguistics
language
university

3 most distinct SPAM words:
credit
check
product
```

- The following is the plot of the classification results of our Naive Bayes implementation:




- $w_0$ in the linear classifier corresponds to the prior in Naive Bayes. It describes a probability which holds regardless of the data point which is classified. $w$ corresponds to the likelihood, since it describes a relation between the data point and the classes. $x$ is thereby the data point.

## 5. Theory/Coding questions LogReg

- | **Number of Misclassifications** | RegVal 0 | RegVal 0.1 | RegVal 1.0 |
  |---|---|---|---|
  | Training | 0 / 38 | 0 / 38 | 0 / 38 |
  | Test | 0 / 37 | 0 / 37 | 2 / 37 |

- As the value of the regularization coefficient increases, the probability lines move away from each other.
- See figures below
- The regularization term $1/(2 * \sigma^2) * ||w||^2$ from the slides is subtracted from the cost function term. Also, the first entry $w_0$ is ignored in this calculation, since the bias term should not influence the result.
- Increasing the regularization term reduces the height (y-axis) and at the same time broadens on the x-axis. This effectively results in a reduction of the weights of the samples.
- The number of misclassified digits for the Zip-13 data set can be read in the following output:

```
##########-##########-##########
LOGREG exercise - MNIST Example 1 vs 3
##########-##########-##########
Dataset ballance in training 60.43%
Dataset ballance in test 61.40%
Training a LOGREG classifier with regularization coefficient: 0.0
initial posteriorloglikelihood -1152.7037612711395 initial likelihood 0.0
final posteriorloglikelihood -1408.625708579227 final likelihood 0.0
Training
85/1663 misclassified. Total error: 5.11%.
Test
43/430 misclassified. Total error: 10.00%.
Training a LOGREG classifier with regularization coefficient: 0.1
initial posteriorloglikelihood -1152.7037612711395 initial likelihood 0.0
final posteriorloglikelihood -53.94461412833775 final likelihood 3.73380605510187e-24
Training
0/1663 misclassified. Total error: 0.00%.
Test
4/430 misclassified. Total error: 0.93%.
Training a LOGREG classifier with regularization coefficient: 1.0
initial posteriorloglikelihood -1152.7037612711395 initial likelihood 0.0
final posteriorloglikelihood -97.40871354853567 final likelihood 4.965159640777318e-43
Training
0/1663 misclassified. Total error: 0.00%.
Test
4/430 misclassified. Total error: 0.93%.
##########-##########-##########
```

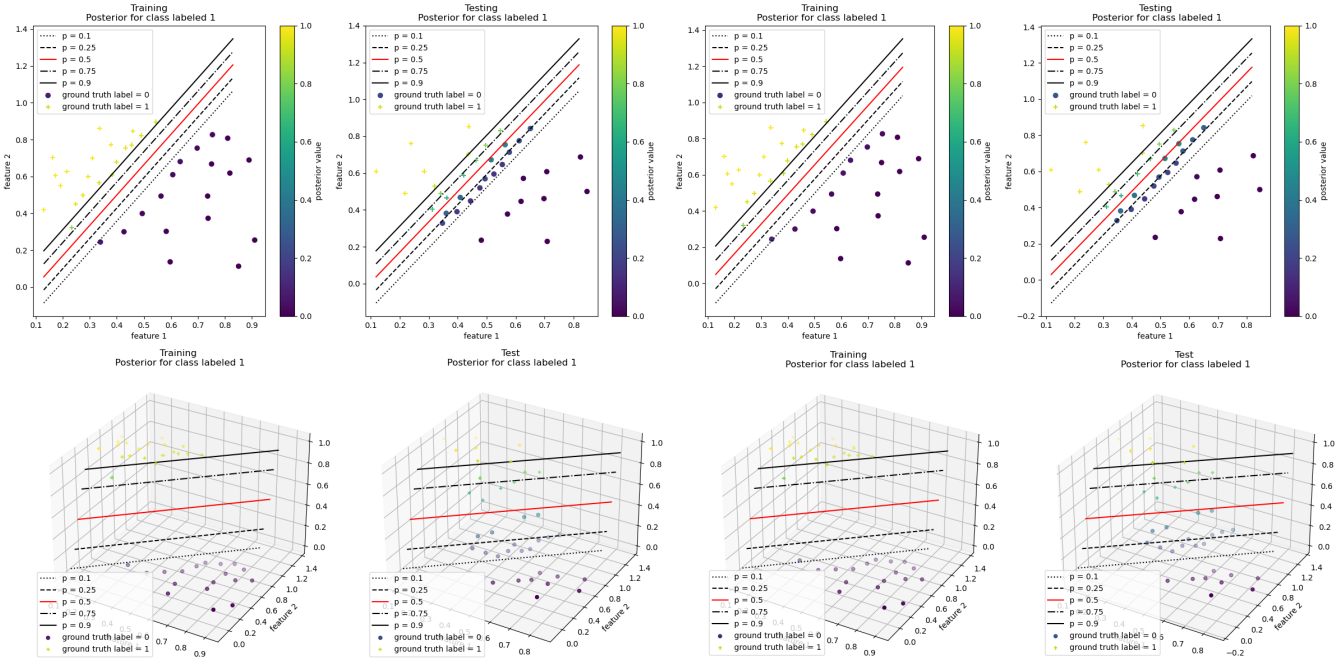



Figure 1: RegVal 0

Figure 2: RegVal 0.1

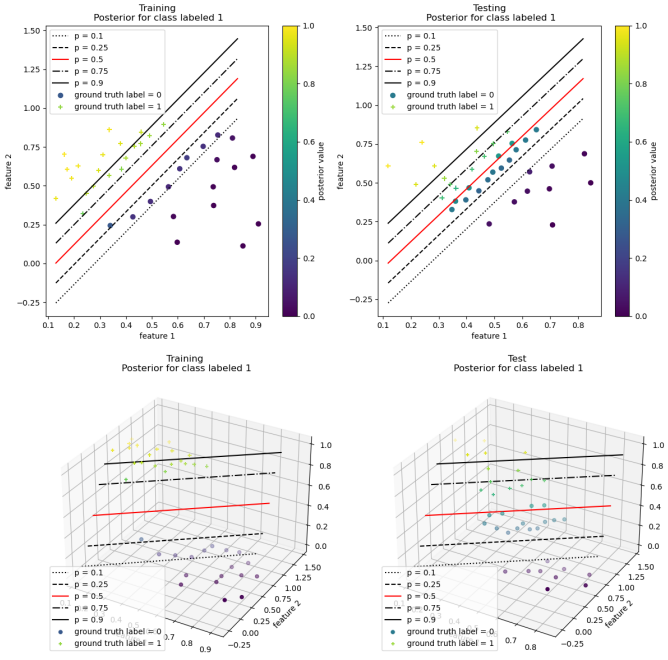



Figure 3: RegVal 1