Assignment 1

Vasco Lage de Oliveira

November 27, 2018

Contents

1 Make Your Own

1. To answer this question, one must understand that there are many ways of approaching this problem. Therefore, I will mention some examples. First, we could consider our sample space as the average of the prior grades achieved by each student during is study program. In this case, our sample space $\mathcal{X}$ would be, $\mathcal{X} = [-3, 12]$. Instead of considering only the average, we can consider his/her study program. As an example of our course, with so many students from different areas, we can consider the label space for this attribute as being the natural numbers. Therefore $\mathcal{X} = [-3, 12] \times \mathbb{N}$.

2. No matter which sample space, $\mathcal{X}$, we have. Our label space will always be the final grades in Machine Learning,i.e., $\mathcal{Y} = \{-3, 0, 2, 4, 7, 10, 12\}$, no matter how we choose $\mathcal{X}$.

3. In this case, predicting $y' = 3$ when $y = 12$, it would be an enormous mistake whilst $y' = 10$ when $y = 12$ would not. Hence zero-one loss function is definitely not advisable for this situation. We could choose the absolute loss function for this problem.

4. I would evaluate the the performance of algorithm in terms of the square loss function in this way:
   If $|y' - y| \leq 3$ then the algorithm was with good performance, otherwise it would have a bad performance.

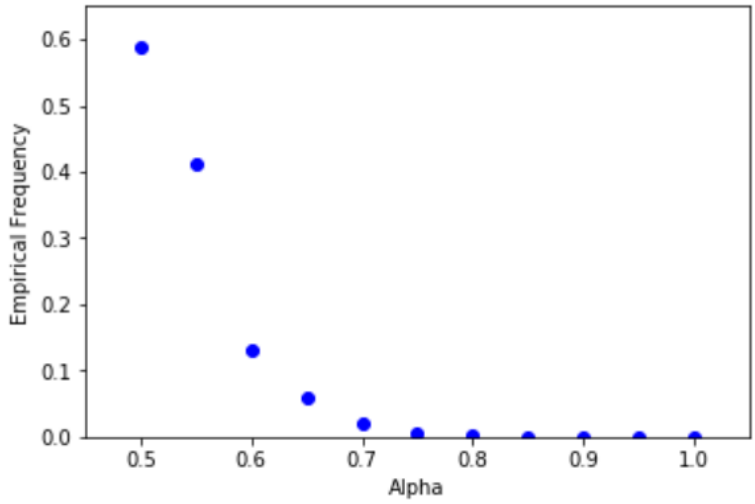
5. Once we started our algorithm we could expect issues coming up such as students who are doing Machine Learning as their first course in their study program, that would mean they would not have any average, thus we would have a conflict dividing by zero (number of courses done before). We could handle this issue by assigning the average of all students in Machine learning on the year before to his/her average.

2 Illustration of Markov's and Chebychev's Inequalities

Note: See ***Exercise_2.zip***

1. In this exercise notice that if $X_1, ..., X_{20}$ are i.i.d Bernoulli random variables with parameter $\frac{1}{2}$, then $\sum_1^{20} X_i$ is a binomial random variable, $Bin(20, \frac{1}{2})$. Using this command in python: **np.random.binomial(n=20,p=0.5,size=1000000)** We make 1,000,000 repetitions of the desired experiment.

2.



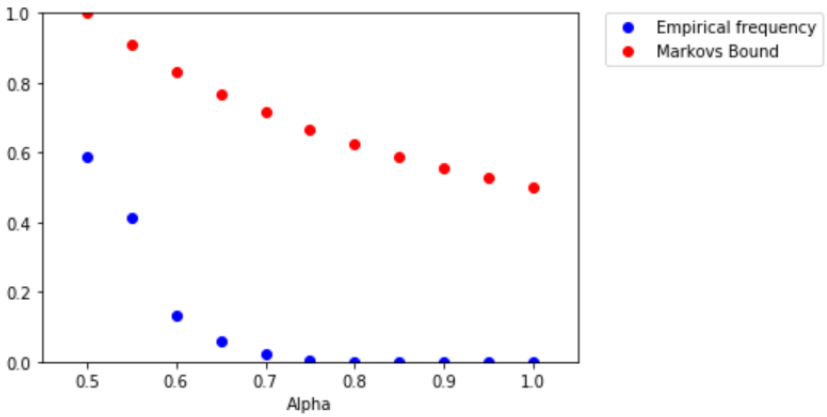
3. Notice that

$$\{(x_1, x_2, ..., x_{20}) : \frac{1}{20}\sum_1^{20} x_i \geq \alpha{+}0.05\} \subseteq \{(x_1, x_2, ..., x_{20}) : \frac{1}{20}\sum_1^{20} x_i \geq \alpha{+}0.01\}$$

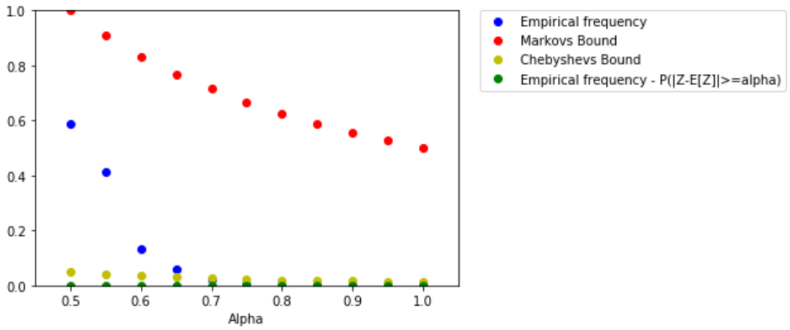
$$\{(x_1, x_2, ..., x_{20}) : \frac{1}{20}\sum_1^{20} x_i \geq \alpha + 0.01\} \subseteq \{(x_1, x_2, ..., x_{20}) : \frac{1}{20}\sum_1^{20} x_i \geq \alpha\}$$

Which means that this function decreases, and therefore 0.05 is enough to see its behaviour.

4. First, notice that $Y = \sum_1^{20} X_i$ is a binomial with $\mathbb{E}[Y] = 20 \times \frac{1}{2} = 10$ and $\mathbb{E}[\frac{1}{20}Y] = \frac{10}{20} = 0.5$. Therefore we have the following plot:



5. In this exercise have in mind that $Var[\frac{1}{20}Y] = \frac{1}{20^2}Var[Y] = \frac{5}{20^2} = 0.0125$. Let $Z = \frac{1}{20}Y$, therefore we have the following plot:



6. We can see in the plots shown above that Markov's bound is loose when bounding the empirical frequency whilst Chebyshev's bound is really tight when bounding $P(|Z - \mathbb{E}(X)|) > \alpha$.

7. For $\alpha = 1$ we want $P(Y \geq 20) = P(Y = 20) = \binom{20}{20}\frac{1}{2^{19}}\frac{1}{2^1} = \frac{1}{2^{20}}$ and for $\alpha = 0.95$ we want $P(Y \geq 20 \times 0.95) = P(Y \geq 19) = P(Y = 20) + P(Y = 19)$. Thus, $P(Y \geq 20 \times 0.95) = \frac{1}{2^{20}} + 20\frac{1}{2^{20}} = \frac{21}{2^{20}}$

3 Tightness of Markov's Inequality

Defining the random variable $Y_{\epsilon^*} = X - \epsilon^* 1_{\{X \geq \epsilon^*\}}$, where

$$1_{\{X \geq \epsilon^*\}} = \begin{cases} 1, \text{ if } X \geqslant \epsilon^* \\ 0, \text{ if } X < \epsilon^* \end{cases}.$$

Observe that Y_ϵ^* is non-negative. Taking the expectation yields

$$E(Y_\epsilon^*) = E(X) - \epsilon^* P(X \geq \epsilon^*).$$

Hence Markov's inequality holds with equality if and only if $E(Y_\epsilon^*) = 0$. Since Y_ϵ^* is non-negative, this is equivalent to $P(Y_{\epsilon^*} = 0) = 1$. Note that $Y_{\epsilon^*} = 0$ if and only if $X = 0$ or $X = \epsilon^*$.

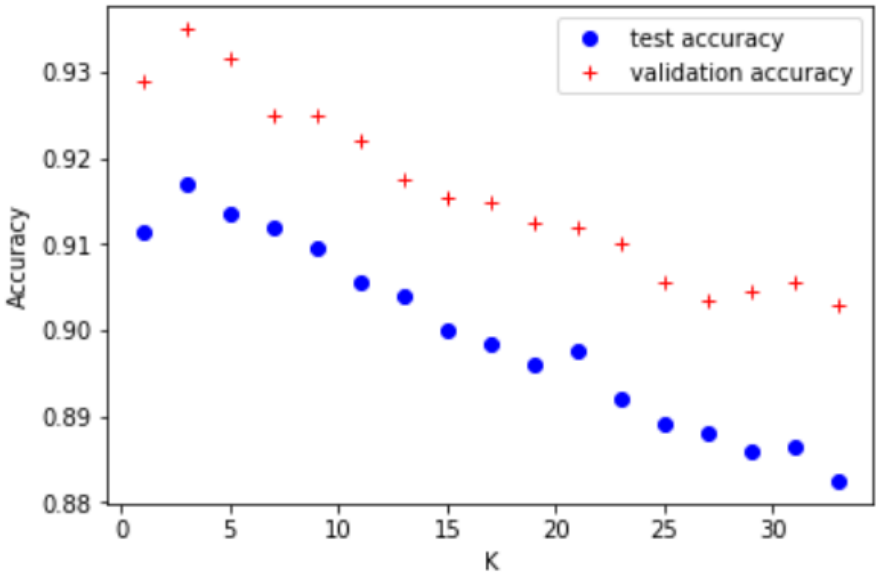
Therefore, Markov's inequality holds with equality if and only if $P(X \in \{0, \epsilon^*\}) = 1$.

4 Digits Classification with Nearest Neighbours

This exercise was solved in the following way:

Note: See ***K_NN.zip***

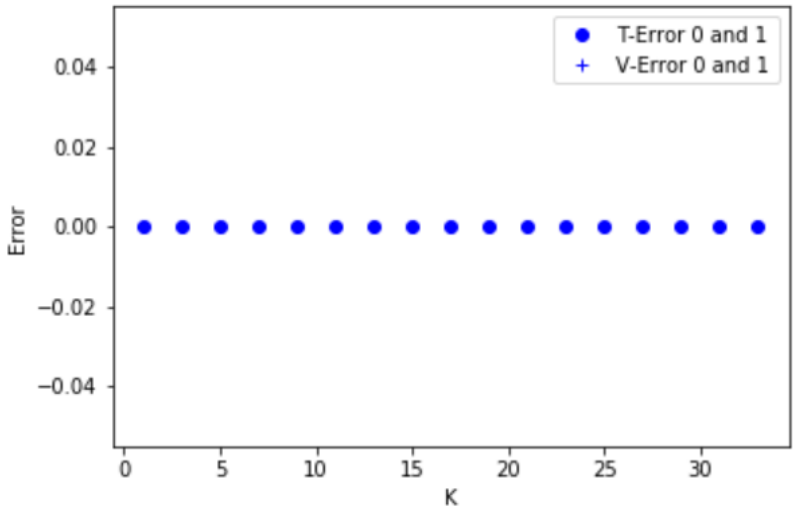
1. I have created a function, *k_nn*, that computes a general algorithm of $K - NN$, and returns a big $\#testsamples \times 17$ - matrix with all the predictions for different K's in $\{1, 3, 5, ..., 33\}$. The function also returns the accuracy of our predictions. This algorithm took one or two minutes two compute when the input was our test set or our validation set. The code for this exercise is the jupyter notebook file **"K_NN.ipynb"** For the validation set and test set I got the following accuracy:



   For both sets, $K = 3$ is the best value of K, and the error behaves as a linear function of K.
   The validation error is approximately 2% less than the test error for every K.
   Now, we are going to do the same analysis for every task.

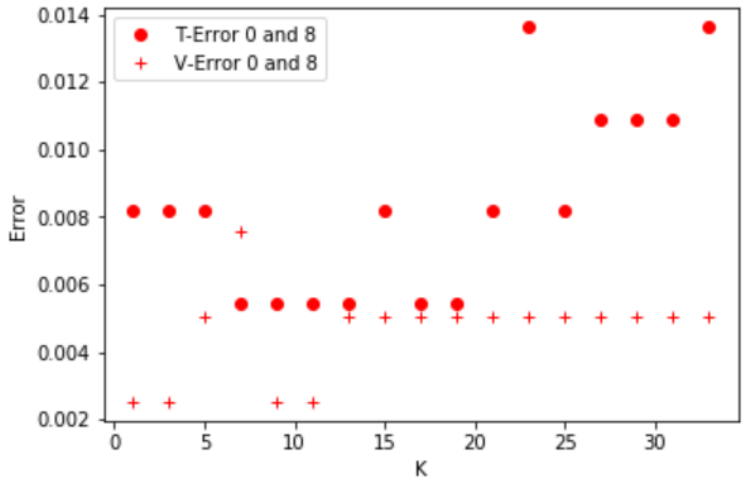
2. For these tasks, I have created a function, *distinguish*, that returns the percentage error of predicting **a** when the real value is **b** and vice-versa, for a given **a**,**b** and K.

(a) Task 1 - Ability to distinguish between 0 and 1.



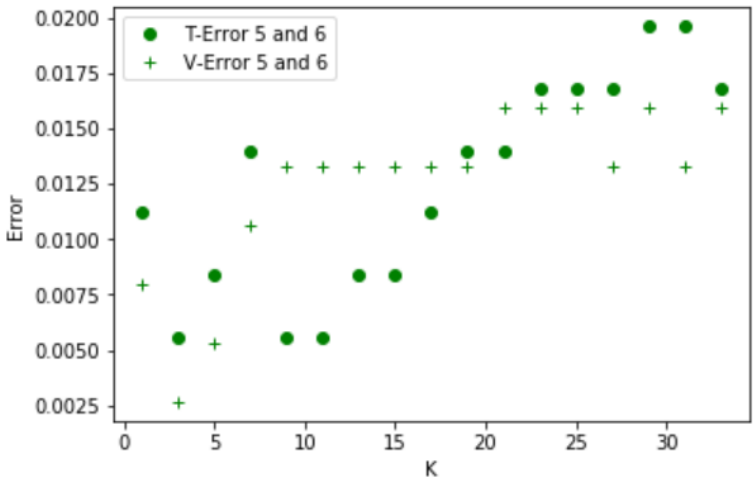
For this task, the algorithm had no problem to distinguish between 0 and 1, for both sets, validation and test, and for all K.

(b) Task 2 - Ability to distinguish between 0 and 8.



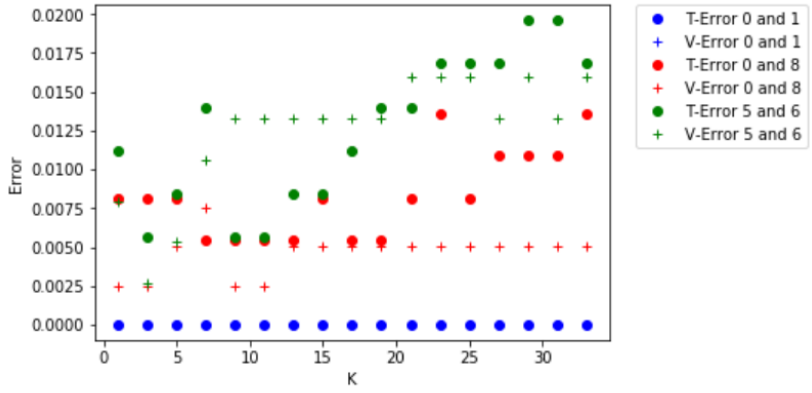
In this case, we can say that the validation error doesn't match the test error for every K but they are close from each other.
The best value of K for the test set is $K = 7, 9, 11, 13, 17, 19$ with approximately 0.5% error, and for the validation set the best value of K is $K = 1, 3, 9, 11$ with approximately 0.3% error.
The validation error is constant for $K \geq 11$, and the test error seems to become larger as K increases.

(c) Task 3 - Ability to distinguish between 5 and 6.



In this last case, we can say that the validation error doesn't match the test error for every K but they are very close from each other. The best value of K for the test set is $K = 3, 9, 11$ with approximately 0.6% error, and for the validation set the best value of K is $K = 3$ with approximately 0.3% error.
The validation error and test error tend to increase as K gets larger.

(d) Finally, when we plot all the errors we get:



We can conclude that is easier to tell apart 0 and 1 than 5 and 6 and, moreover, the difficulty of separating 0 and 8 is between the other tasks. However, the best value of K does not depend on difficulty of the task.

5 Nearest Neighbours for Multiclass Classification

Note: See ***K_NN.zip***

This is exercise was somehow done in the previous one. I have chosen the mode of the K-nearest neighbors to determine Y', in other words, for all labels we can count all the K-nearest neighbors that have the corresponding label, the label with higher value is Y'. The algorithm is available in the jupyter notebook file **"K_NN.ipynb"**.

6 Linear Regression

1. See `Linear_Regression.zip`
2. The affine linear model using linear regression on the provided data is:

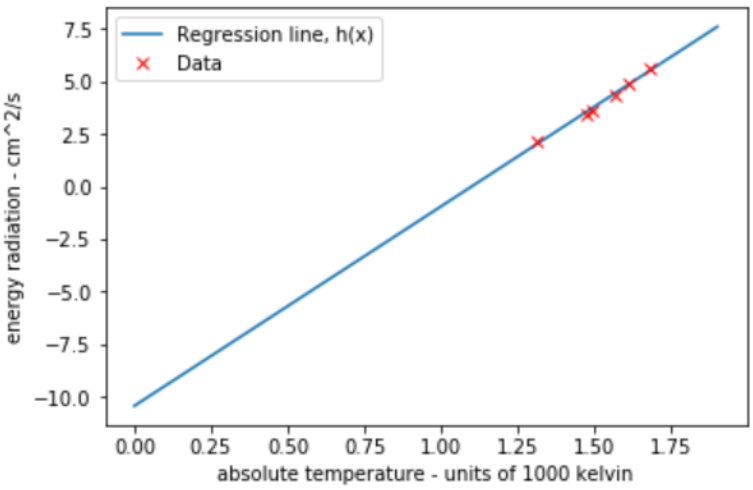
$$h(x) = ax + b,$$

with $a = 9.49$ and $b = -10.43$. Therefore,

$$h(x) = 9.49x - 10.43.$$

The mean squared error given by, $MSE = \frac{1}{6}\sum_i (y_i - h(x_i))^2 = 0.01$

3. We have the following figure:



4. The variance of $\mathbf{y}$ is given by $Var(\mathbf{y}) = \frac{1}{6}\sum_i (y_i - \overline{\mathbf{y}})^2$ where $\overline{\mathbf{y}}$ is the mean of $\mathbf{y}$. We have $Var(y) = 1.27$.
The MSE is much smaller than the $Var(y)$, which means that we have a $R^2 = 1 - \frac{MSE}{Var(Y)}$ really close too one, because $\frac{MSE}{Var(Y)} = 0.01$, i.e., the model is very good. $\frac{MSE}{Var(Y)}$ being larger than one means that the model

does not explain any of the variation in the response variable around its mean whilst being smaller than one means that the model explains all of the variation in the response variable around its mean.

5. In this model, we have $h(x) = 1.42x^3 - 1.07$. The mean squared error, MSE, is, $MSE = 0.0005$, extremely low, almost zero. As it becomes clear when we observe the plot: