Problem 1. (Train, validation, test [15 points])

You are addressing a regression problem with $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. You have tried five different approaches: A, B, C, D, E. Each approach gives you a predictor. So, your set of predictors are $f_A, f_B, f_C, f_D, f_E : \mathbb{R}^d \to \mathbb{R}$. You divided the dataset into a training and test set. You further performed 5-fold cross-validation on the training set. You obtained the following average train error and validation error (averaged over the 5-folds) for each model.

model	train error	validation error
A	1.355	1.423
B	9.760	9.165
C	5.033	0.889
D	0.211	5.072
E	0.633	0.634

Answer the following questions. Note that full marks will be given only if you justify your answer.

1. Let $I_1, I_2, \ldots, I_5$ denote the indices of the datasets in the validation set for each fold. For model A, write the formula for the average and the standard deviation of the mean-square error of the 5 folds.

   Solution: The average of the mean-squared error (MSE) of the 5 folds of model A is defined as follows:

   $$\overline{\text{MSE}}_A = \frac{1}{5}\sum_{i=1}^{5} \text{MSE}_A^i = \frac{1}{5}\sum_{i=1}^{5} \frac{1}{|I_i|} \sum_{(x,y)\in I_i} (f_A(x) - y)^2$$

   The standard deviation of the MSE of the 5 folds of model A is defined as follows:

   $$\sigma_A = \sqrt{\frac{1}{4}\sum_{i=1}^{5} (\text{MSE}_A^i - \overline{\text{MSE}}_A)^2},$$

   where

   $$\text{MSE}_A^i = \frac{1}{|I_i|} \sum_{(x,y)\in I_i} (f_A(x) - y)^2. \tag{1.1}$$

   Clarification: The reason why we use 4 instead of 5 in the denominator for computing standard deviation is because $\overline{\text{MSE}}_A$ is a sample mean and we want to get unbiased estimator. The same reason applied in question 6 in problem 4 and question 4 in problem 2. See more details in Clarifications on Homework 3 in moodle.

2. Which model(s) seems to be overfitting?

   Solution: Model D has a very low training error but at the same time, it has a large validation error. Therefore, Model D seems to be overfitting.

3. Which model(s) seems to be underfitting?

   Solution: Model B and Model C both have a large training error. It seems both models do not fit the data well and are therefore underfitting.

4. Your colleague tells you that he does not believe the result of one of your models and he thinks you might have made a mistake in coding one of the models. Based on the train and validation errors above, which model is more likely to have a mistake in its coding?

   Solution: Model C has a very low validation error compared to its relatively high training error. This can be an indication that there is an error in the code.

5. Suppose that the variance of the error across the folds for model E is very high and for model B is much lower. Which model is likely to have a test error on unseen data more similar to the validation error reported above?

   Solution: Model B is more likely than Model E to have a test error on unseen data that is similar to the reported validation error because the variance of Model B is much lower compared to Model E. Recall that the variance of a data set captures the spread of the data set around its mean.

6. Your friend suggests that you average model A and model B outputs for the regression task. Explain how you could check the correlation of the errors of the models to determine whether this averaging could potentially give you a better result on unseen data?

   Solution: If the errors of Model A and Model B are uncorrelated, then averaging the outputs of the two models can give better results on unseen data. To check whether the errors of Model A and Model B are uncorrelated, one can compute the correlation of the MSE of model A and the MSE of Model B:

$$\text{corr}(\text{MSE}_A, \text{MSE}_B) = \frac{cov(\text{MSE}_A, \text{MSE}_B)}{\sqrt{cov(\text{MSE}_A, \text{MSE}_A)}\sqrt{cov(\text{MSE}_B, \text{MSE}_B)}},$$

   where the covariance is computed as $cov(\text{MSE}_k, \text{MSE}_j) = \frac{1}{4}\sum_{i=1}^{5}(\text{MSE}_k^i - \overline{\text{MSE}}_k)(\text{MSE}_j^i - \overline{\text{MSE}}_j)$ and $k, j \in \{A, B\}$ (See Equation (1.1) for the defintion of the MSE).

7. Suppose that model A corresponds to a decision-tree with depth 2 whereas Model E corresponds to a neural network with 5 hidden layers. Which of the two models would you choose for the regression problem and why? You might consider that the regression task impacts humans and interpretability is helpful.

   Solution: When humans are involved and interpretability is desired, neural networks are usually a bad choice, since they might take uninterpretable and inexplicable decisions. Decision-trees, on the other hand, provide the user with a clear decision path such that the user can follow what she is doing at each step to explain why a particular answer is chosen. Note that if the depth of the tree is too large, then interpreting a decision tree can also be difficult.

Problem 2. (Naive Bayes classifier [22 points])

We consider a cancer diagnosis problem. In this problem, our features are $x \in \mathbb{R}^2$, with x_1 denoting the average radius and x_2 denoting the average texture of a tumor. The classification is whether the tumor is *malignant* or *benign* based on the feature x[1]. For N number of patients a medical expert has labeled the data x^i with $y^i = 0$ for benign and $y^i = 1$ for malignant tumor. Our goal is to design an algorithm that learns to do the labelling for a new patient by measuring the x_1, x_2 of its tumor. For this, we want to use a Naive Bayes Classifier.

1. Given a data point x write the Bayes rule for determining the conditional probability of class 0 and class 1.

   Solution. $P(y = C|x) = \frac{pdf(x|y=C)P(y=C)}{pdf(x)}$, where $pdf(x)$ denotes the probability density function of x and $pdf(x|y = C)$ denotes the conditional probability density of x given class C, where $C \in \{0, 1\}$.

2. Suppose our dataset contained feature and diagnosis for $N = 1000$ patients, from which 50 had a malignant tumor. What is the empirical estimate of prior probability $P(y = k)$ for $k \in \{0, 1\}$ based on this data?

   Solution. $P(y = 1) = 50/1000 = 5\%$, $P(y = 0) = 950/1000 = 95\%$

3. Suppose we use a Gaussian Naive Bayes Classifier in the rest of this exercise. Write the assumptions underlying this model.

   Solution. First, the Naive Bayes assumption is that the features x_1 and x_2 are independent conditioned on the class C. This is written as $pdf(x_1, x_2|y = C) = pdf(x_1|y = C)pdf(x_2|y = C)$. Second, the Gaussian assumption is that $pdf(x_i|y = k) = \mathcal{N}(\mu_{i,k}, \sigma^2_{i,k})$, for $i = 1, 2$ and $k = 0, 1$.

4. Explain how you would compute the conditional distribution of average radius of a tuomr, conditioned on the tumor being malignant.

   Solution. We will consider the data points with label class malignant. Let us denote their indices by $N_1 \subset \{1, 2, \ldots, N\}$. To model the Gaussian distribution of tumor size conditioned on tumor being malignant, we compute the mean $\mu_{1,1} := \frac{1}{|N_1|} \sum_{i \in N_1} x^i_1$ and standard deviation of these points $\sigma_{1,1} := \sqrt{\frac{1}{|N_1|-1} \sum_{i \in N_1} (x^i_1 - \mu_{1,1})^2}$. Then, we set $pdf(x_1|y = 1) = \mathcal{N}(\mu_{1,1}, \sigma^2_{1,1})$.

5. After fitting the parameters of the Gaussian distribution for each feature and conditioned on each type of tumor, you found that your classifier classifies only 40 of the 50 malignant tumors as malignant and it classifies 945 of the benign tumors as benign. Let the malignant tumor correspond to the "positive" class. What is the number of false positives, the number of false negatives and the error rate of your classifier?

   Solution. The number of false positives: 5 (5 of benign tumors are classified as malignant), number of false negatives: 10 (only 40 out of 50 malignant tumors classified correctly), error rate $= \frac{5+10}{1000} = 1.5\%$.

6. Given a new patient's feature data x, write the formula for determining whether the patient has a malignant or a benign tumor based on your trained classifier.

[1] A tumor is an abnormal collection of cells. It forms when cells multiply more than they should or when cells don't die when they should. A tumor can be malignant (cancerous) or benign (not cancerous)source

Solution. We declare the patient has malignant tumor if $P(y=1|x) > P(y=0|x)$. Now, to do the above comparison, we use the Bayes rule and the Naive Bayes assumption to arrive at

$$
\begin{aligned}
&P(y=1|x) > P(y=0|x) \iff \\
&\frac{pdf(x|y=1)P(y=1)}{pdf(x)} > \frac{pdf(x|y=0)P(y=0)}{pdf(x)} \iff \\
&pdf(x|y=1)P(y=1) > pdf(x|y=0)P(y=0) \iff \\
&pdf\left(x_1|y=1\right) pdf\left(x_2|y=1\right) P\left(y=1\right) > pdf\left(x_1|y=0\right) pdf\left(x_2|y=0\right) P\left(y=0\right) \iff \\
&\log\left(pdf\left(x_1|y=1\right) pdf\left(x_2|y=1\right) P\left(y=1\right)\right) > \log\left(pdf\left(x_1|y=0\right) pdf\left(x_2|y=0\right) P\left(y=0\right)\right) \iff \\
&\log(pdf\left(x_1|y=1\right) + \log pdf\left(x_2|y=1\right) + \log P\left(y=1\right) \\
&> \log pdf\left(x_1|y=0\right) + \log pdf\left(x_2|y=0\right) + \log P\left(y=0\right) \iff \\
&\log\left(\frac{1}{\sqrt{2\pi}\sigma_{1,1}} e^{-\frac{\left(x-\mu_{1,1}\right)^2}{2\sigma_{1,1}^2}}\right) + \log\left(\frac{1}{\sqrt{2\pi}\sigma_{2,1}} e^{-\frac{\left(x-\mu_{2,1}\right)^2}{2\sigma_{2,1}^2}}\right) + \log P\left(y=1\right) \\
&> \log\left(\frac{1}{\sqrt{2\pi}\sigma_{1,0}} e^{-\frac{\left(x-\mu_{1,0}\right)^2}{2\sigma_{1,0}^2}}\right) + \log\left(\frac{1}{\sqrt{2\pi}\sigma_{2,0}} e^{-\frac{\left(x-\mu_{2,0}\right)^2}{2\sigma_{2,0}^2}}\right) + \log P\left(y=0\right) \iff \\
&\log\frac{\sigma_{1,0}\sigma_{2,0}}{\sigma_{1,1}\sigma_{2,1}} + \frac{1}{2}\left(\frac{\left(x-\mu_{1,0}\right)^2}{\sigma_{1,0}^2} + \frac{\left(x-\mu_{2,0}\right)^2}{\sigma_{2,0}^2} - \frac{\left(x-\mu_{1,1}\right)^2}{\sigma_{1,1}^2} - \frac{\left(x-\mu_{2,1}\right)^2}{\sigma_{2,1}^2}\right) > \log\frac{P\left(y=0\right)}{P\left(y=1\right)}
\end{aligned}
$$

7. Suppose you now use your classifier to analyse data from new patients. The medical doctor asks you to bring the cases for which the machine learning has less confidence regarding the diagnosis directly to him so that he can use his expert knowledge. How would you use the information obtained from the probabilistic prediction of the Naive Bayes filter to decide which cases need additional attention?

   Solution. We should ask the expert to check the patients for which probability of the two classes is close to each other based on our classifier. The reason is that if the probability of malignant and benign are close, and hence, both near 0.5, we cannot be very confident in the diagonais of our algorithm.

Problem 3. (Neural networks [15 points])

We have an audio signal from a piece of music and we want to classify the music according to its genre[2]. For this, we use a training dataset which consists of a library of audio signals, labelled according to their genre. Let $x \in \mathbb{R}^d$ denote an audio signal. Here, $x = (x_1, x_2, \ldots, x_d)$ with x_i denoting the acoustic pressure measured at time step i. For training, we use a dataset consisting of 1,000 4-second music excerpts evenly distributed into nine classes: rock, reggae, blues, classical, disco, country, hip-hop, jazz, and pop. We consider audio samples with 5,000 samples per second for 4-second. Therefore, the input to our classifier is a $5,000 \cdot 4 = 20,000$-dimensional vector.

1. Suppose we have a neural network with two hidden layers and an output layer for the 9 classes. Each hidden layer has 10 nodes. How many weights and biases need to be determined for each layer? Show your work.

   Solution. There are $20,000 \cdot 10$ weights in the first layer, $10 \cdot 10$ weights in the second layer, and $10 \cdot 9$ in the last layer. As for the biases, there are 10 biases in the first layer, 10 biases in the second layer, and 9 biases in the last layer. In total there are $200,000 + 100 + 90 + 10 + 10 + 9 = 200,219$ weights and biases.

2. You observe that the number of training data you have is relatively small compared to the number of parameters. After training the network, you get a very small training error. Hence, you suspect your neural network is possibly overfitting to the training data. What approach could you use to reduce this potential overfit?

   Solution. We can add a regularization term (1-norm or 2-norm penalty) to the loss. Another approach, which has not been discussed in the course, would be to randomly discard a certain percentage of nodes during training. This is called dropout. Note that you are not required to know about dropout for the exam but you can read about it for your future if you wish.

3. Your friend who is very musical thinks that music audio signals can be distinguished based on local characteristics of the signal and she suggests you to use a 1-dimensional convolutional neural network. Suppose now for each of the first and second layer of the neural network, you use 128 filters of dimension 5 for each layer. Hence, each filter is given by $w = (w_1, w_2, \ldots, w_5)$, where w_i's needs to be designed. The filter is applied with a stride of 5. It follows that after applying each filter to an audio signal of length d, we will have a signal of length $\frac{d}{5}$. How many parameters need to be determined for the first, the second and the output layer? Show your work.

   Solution. Denoting by $C_{\text{in}}^{[j]}$ the number of input channels, by $F^{[j]}$ the kernel size of the filter, and by $C_{\text{out}}^{[j]}$ the number of filters of a 1-dimensional convolutional layer j, there are $C_{\text{in}}^{[j]} \times F^{[j]} \times C_{\text{out}}^{[j]}$ weights and $C_{\text{out}}^{[j]}$ biases in this layer. Furthermore, when having multiple convolutional layers it holds $C_{\text{in}}^{[j+1]} = C_{\text{out}}^{[j]}$, since we get for each filter in layer j a channel in layer $j+1$. Thus, our convolutional neural network has $1 \cdot 5 \cdot 128 + 128 = 768$ parameters in the first convolutional layer and $128 \cdot 5 \cdot 128 + 128 = 82,048$ parameters in the second convolutional layer. Due to the stride of 5, the first hidden layer consists of 128 channels with each channel having $20,000/5 = 4,000$ nodes. Similarly, the second hidden layer consists of 128 channels with each channel having $4,000/5 = 800$ nodes. Thus, after flattening the second hidden layer we get a length $128 \cdot 800 = 102,400$ vector and the last fully connected layer has $102,400 \cdot 9 + 9 = 921,609$ parameters. Comparing this to the $200,219$ parameters in Problem 3.1, we observe that this convolutional neural network has more trainable parameters.

[2]Such a problem arises in music streaming or music shopping services.

However, as opposed to the fully connected neural network it has $128 \cdot 4000$ nodes (instead of 100) in the first hidden layer and $128 \cdot 800$ nodes (instead of 100) in the second hidden layer. A fully connected neural network with the same number of nodes in each hidden layer would have considerably more parameters than the convolutional neural network.

Remark: The size of the second hidden layer results in a large weight matrix for the final fully connected layer. In practice, the input data is usually further compressed by additional convolutional layers and a pooling operation before the fully connected layer(s). This can result in the CNN to have lower number of parameters compared to a fully connected network.

4. You train the network above using stochastic gradient descent, with a batch size of 100 and for 10 epochs. How many iterations of gradient descent is being run in each epoch?

   Solution. Since (i) each epoch runs through the entire data set, (ii) we have a sample of 1000 data points, (iii) the batch size, namely, the number of samples used in each iteration is 100, we get $1000/100 = 10$ gradient descent iterations per epoch.

5. Your friend says that she has used the same network architecture, learning rate, and batch size. However, her training and test error are different from yours. What could be some reasons for the difference?

   Solution. She can either have trained for a different number of epochs, or it is the effect of randomness. Such randomness arises in initialization of the weights and in assignment of the samples to the batches.

Remark: For more information regarding this approach, you may see this paper, and for the audio dataset you may see here.

Problem 4. (Decision trees [18 points])
Consider a classification problem with $x \in \mathbb{R}^2$ and $y \in \{\text{square}, \text{triangle}\}$. The training data is shown in Figure 1 below. There are N_t triangles and N_s squares in the training data, where $N_s = mN_t$ with $m \in (0, 1)$. So, for example, if there are 100 triangles, and $m = .1$. then there are 10 red squares and a total of 110 data points.

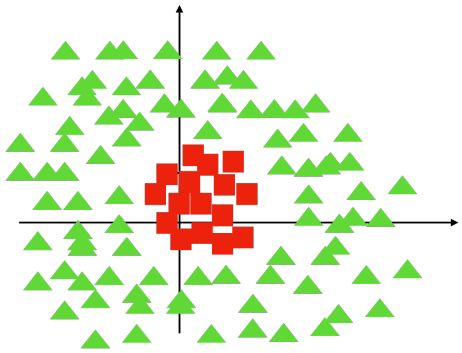

Figure 1: Classification problem training data

1. A so-called null classifier gives the majority label of the training data to any test point $x \in \mathbb{R}^2$. Hence, it considers that x has no effect on the label. Since we have $N_t = (1/m)N_s > N_s$ the majority label is triangle and the null-classifier labels any test point x as a triangle. What is the gini index of this classifier? What is the error rate of this classifier on the training data?

   Solution. The probabiThe gini index is $m/(1+m)*1/(1+m)+1/(1+m)*m/(1+m) = \frac{2m}{(1+m)^2}$. Since the classifier gets all the squares wrong, its error rate is $\frac{mN_t}{(1+m)N_t} = \frac{m}{1+m}$.

2. Now, consider feature 1 and the threshold at $x_1 = 1$ shown in Figure 2 below as a candidate for forming a split in a first node of a decision tree to be constructed for classification. So, the split criteria is whether $x_1 > 1$. Suppose that a fraction of $c \in (0, 1)$ number of triangles falls to the right of the line at $x_1 = 1$ shown in the figure. In other words, cN_t of triangles have $x_1 > 1$. Hence, $(1 - c)N_t$ are the number of triangles to the left of the line. Write the gini index of the two leaves and of the node according to this split.

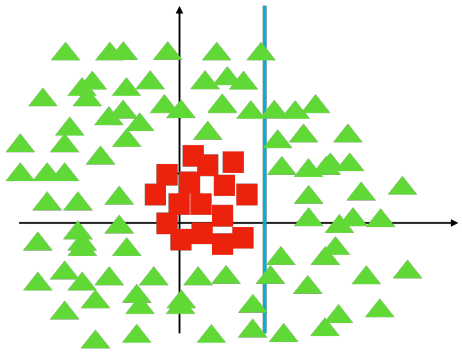

Figure 2: Classification problem with one node of the decision tree

   Solution. The gini index of the leaf resulting from satisfaction of $x_1 > 1$ is 0. This is because the leaf is pure and all the data points are identified correctly as a triangle. The gini index of the leaf resulting from NOT satisfying $x_1 > 1$ is as follows.

First, there are $(1-c)N_t$ triangles and mN_t squares for a total of $(1-c+m)N_t$ data points with $x_1 \leq 1$. Within these data points:
Probability of class triangle is $(1-c)N_t/(1-c+m)N_t = \frac{1-c}{1-c+m}$.
Probability (fraction) of class square is $mN_t/(1-c+m)N_t = \frac{m}{1-c+m}$.
Finally, gini of this leaf is $\frac{1-c}{1-c+m} * \frac{m}{1-c+m} + \frac{m}{1-c+m} * \frac{1-c}{1-c+m} = \frac{2(1-c)m}{(1-c+m)^2}$.

It follows that the gini index of the node with this split is

$$0 \times \frac{c}{1+m} + \frac{2(1-c)m}{(1-c+m)^2} \times \frac{(1-c+m)}{1+m} = \frac{2(1-c)m}{(1+m)(1-c+m)}.$$

3. Show that the gini index after the split is smaller than the gini index of the null classifier.

   Solution. We need to compare the gini index after the split $\frac{2(1-c)m}{(1+m)(1-c+m)}$ to that before the split $\frac{2m}{(1+m)^2}$. In particular, we should show that $\frac{2m(1-c)}{(1-c+m)(1+m)} < \frac{2m}{(1+m)^2}$. Now, note that:

$$\begin{aligned}
& m > m(1-c) \quad \text{since } 1-c \in (0,1) \\
\iff & 1-c+m > 1-c+m(1-c) \text{ by adding } 1-c \text{ to both sides of the inequality} \\
\iff & \frac{1}{1-c+m} < \frac{1}{(1-c)(1+m)} \text{ taking the inverse of above} \\
\iff & \frac{1-c}{1-c+m} < \frac{1}{1+m} \text{ multiplying both sides by } 1-c \\
\iff & \frac{2m(1-c)}{(1-c+m)(1+m)} < \frac{2m}{(1+m)^2}
\end{aligned}$$

   Hence, we arrived at the desired result starting from the fact that $(1-c) \in (0,1)$.

4. Observe that anywhere you put a line, the number of "triangles" is more than the number of squares. Thus, show that no matter where you put the blue line, the accuracy of the classifier does not improve, even though its gini index can improve[3]

   Remark: note that this is the case also if you put the lines horizontally or vertically. In particular, this shows that gini index could be potentially a more useful criteria for forming the threshold than accuracy. Note that the performance of the final classifier is measured in terms of accuracy regardless of the criteria used.

   Solution. In any split, you will end up with more triangles than squares on *both sides*, so that the final decisions (in the leaves) will always be "triangle", and the accuracy will always be $\frac{1}{1+m}$.

5. Draw the boundaries corresponding to a decision tree that could separate the two classes.

   Solution. It will look like a square around the red squares.

[3]This is one of the motivations of using other criteria than accuracy in defining the decision-trees.

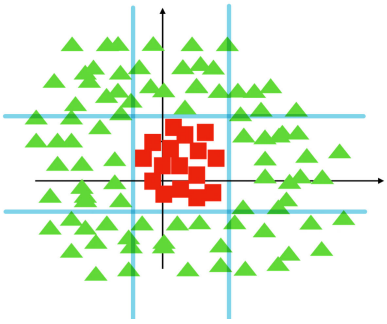

Figure 3: Square boundary corresponding to a decision tree

6. What is the depth of the decision-tree that separates these two classes?

   Solution. It is a tree of depth 4, since you need 4 decision boundaries to carve it.

7. Your friend suggests to you to use a logistic regression for this classification problem. She thinks that it is sufficient to consider two feature as $\Phi_1(x_1, x_2) = x_1^2 + x_2^2$ and $\Phi_2(x_1, x_2) = 1$ for the logistic regression problem. How many parameters you would need to learn for the logistic regression model? What would the decision boundaries look like in this case?

   *Solution.*You would need to learn two parameters, one for each feature, corresponding to a circular decision boundary with center $(0, 0)$. You can construct your predictor as follows:

   (a) You learn the predictor z which can be represented as

   $$z(x_1, x_2) := w_1\Phi_1(x_1, x_2) + w_2\Phi_2(x_1, x_2) = w_1\left(x_1^2 + x_2^2\right) + w_2.$$

   (b) You make your prediction based on $z(x_1, x_2)$ as

   $$\hat{y} = \begin{cases} \text{triangle}, z(x_1, x_2) \geq 0 \\ \text{square}, z(x_1, x_2) < 0. \end{cases} \tag{1.2}$$

   Note that when $z(x_1, x_2) \geq 0$ which is $x_1^2 + x_2^2 \geq -\frac{w_2}{w_1}$, and this requires $w_1 > 0$ and $w_2 < 0$ in this case. A potential solution based on optimizing for the weights above could like like the following picture.

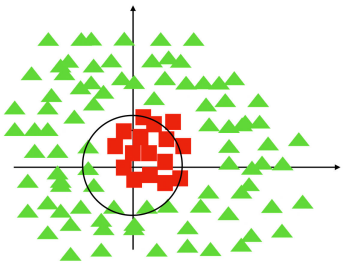

Figure 4: Circular boundary corresponding to logistic regression problem