Problem 1. (KNN, cross-validation [15 points])
A political campaign is on the way for the upcoming election between Party A and Party B in city Z. Party A leader has decided to use machine learning to predict the current trend, based on which it will further adjust its advertisement. To this end, it has surveyed 600 people in the city, recording the coordinate of their home (latitude and longitude), and obtaining their political preference as either A or B. It then has attempted to use the KNN (k-nearest neighbor) approach to predict the potential outcome of the election by estimating the preference for the rest of the residents, having knowledge of the voters' home coordinates.

1. What are the features of the problem? (2 point)

   Solution: There are two features the latitude and longitude of the surveyed people.

2. Consider the problem of predicting political preference of a new individual based on the above data. Is this a supervised or an unsupervised machine learning problem? Justify your answer. (1 point)

   Solution: The problem of predicting the political preference of a new individual is a supervised machine learning problem since labels (A or B) are given for each data point $x^i \in \mathbb{R}^2$, where x_1^i specifies the latitude and x_2^i the longitude of the person's home.

3. Write the formula for standardizing the first feature vector to zero-mean and unit variance. (2 point)

   Solution: The mean and standard deviation of the first feature are given by

   $$\overline{x}_1 = \frac{1}{600}\sum_{i=1}^{600} x_1^i,$$

   and

   $$\sigma_1 = \sqrt{\frac{1}{600}\sum_{i=1}^{600}(x_1^i - \overline{x}_1)^2}.$$

   The formula to standardize the first feature vector to zero-mean and unit variance is then given by:

   $$\hat{x}_1^i = \frac{x_1^i - \overline{x}_1}{\sigma_1}.$$

4. You are given a new person's latitude and longitude (x_1, x_2).

   (a) Write the steps for determining the label of this data point based on $K = 3$ neighbors and using the Euclidean distance (3 point).

   Solution:

   i. Suppose we are given a new data point x. We start by computing the Euclidian distance of this test data point to each training data point (x_1^i, x_2^i) for $i = 1, \ldots, 600$:

   $$dist(x, x^i) = \sqrt{(x_1 - x_1^i)^2 + (x_2 - x_2^i)^2}.$$

ii. We then determine the three training data points with the smallest Euclidian distance to the test data point.

$$j, k, l = \underset{i=1,\ldots,600}{\arg\min}\ dist(x, x^i).$$

iii. Finally we assign the label corresponding to the majority vote of the 3 closest neighbors y^j, y^k, y^l identified in step (ii), where y^i represent the preference for Party A or B of person i.

(b) Specify which step would change and how if you use the Manhattan distance instead of the Euclidean distance. (1 point)

Solution: The only change comes in the way the distance to each point in the dataset is computed in step (i), which now relies on the Manhattan distance:

$$dist(x, x^i) = |x_1 - x_1^i| + |x_2 - x_2^i|.$$

Steps (ii) and (iii) stay the same since they are independent of the distance function used to identify the K nearest neighbors.

Party A wants to choose a good K for its prediction. It thus has divided the data into 500 training points and 100 test points. Then, it has done 5-fold cross validation for the values of $K \in \{1, 3, 5, 7, 9\}$. The following table summarizes the resulting mean and variance of the cross validation error.

model	mean cross-validation error	variance
K=1	0.43	14.23
K=3	0.33	11.50
K=5	0.29	10.41
K=7	0.30	12.23
K=9	0.39	15.44

5. Let $\hat{y}^i$ be the predicted label for data (x_1^i, x_2^i) , y^i its true label, and $\mathbf{1}(\hat{y}^i = y^i)$ be the indicator function. Write the formula for the mean (over the 5 folds) validation error rate for $K = 3$. (2 point)

Solution: The mean cross-validation error $\bar{e}$ can be computed as the average of the error e_f of each of the five folds:

$$\bar{e} = \frac{1}{5}\sum_{f=1}^{5} e_f = \frac{1}{5}\sum_{f=1}^{5}\left(\frac{1}{100}\sum_{i\in I_f}\mathbf{1}(\hat{y}^i = y^i)\right),$$

where I_f gather the indices of the validation points in fold f.

6. For which K is it most likely that the test error will be similar to the validation error, and why? (2 point)

Solution: It is most likely that the model where $K = 5$ has a test error that is similar to the validation error since the variance is the lowest. A lower variance indeed indicates that this model is more robust against different data points being used as training points and hence more likely to perform similarly on some unseen data points.

7. Based on KNN classification, how can Party A know in which region it should improve its advertising campaign? (2 point)

Solution: It is sensible for Party A to increase its advertising campaign in geographic regions where Party B is getting the majority of votes. To that end, Party A can use the proposed KNN model's decision boundaries and overlay them on the city map to find out where Party B is leading the votes.
This of course assumes that the surveyed people are a representative sample of the true data, but it is the best we can do with the given information, and it's certainly not a bad idea to put more effort in advertising where Party B is dominant!

Problem 2. (Feature vectors and linear regression [12 points])
You have recorded the signal corresponding to the mechanical vibration of a turbine for $1, 2, \ldots, T$ time steps as $(y^1, y^2, \ldots, y^T)$, where $y^t \in \mathbb{R}$. You believe that under normal turbine operating conditions, the signal is periodic. In particular, it is approximately the sum of K sinusoidal signals:

$$y^t \approx \sum_{k=1}^{K} a_k \cos(\omega_k t - \phi_k),$$

where ω_k's are known but a_k and ϕ_k are unknown, for $k = 1, 2, \ldots, K$. Based on the observations $\{y^t\}_{t=1}^T$ you want to estimate a_k, ϕ_k so as to minimize the mean square error between the prediction $\hat{y}^t = \sum_{k=1}^K a_k \cos(\omega_k t - \phi_k)$ and the true value y^t.

1. How many parameters do you need to identify for your approximation? (1 point)

   Solution: We need to identify $2K$ parameters: $a_1, \ldots, a_K$ and $\phi_1, \ldots, \phi_K$ to approximate the given signal.

2. Recall the fact that for a sinusoid signal we have:

   $$a\cos(\omega t - \phi) = \alpha\cos(\omega t) + \beta\sin(\omega t),$$

   where $\alpha = a\cos\phi$, $\beta = a\sin(\phi)$. Based on this fact, formulate your predictor $\hat{y} = \sum_{k=1}^K a_k \cos(\omega_k t - \phi_k)$ as a linear function of the parameters α_k, β_k, $k = 1, 2, \ldots, K$, corresponding to each sinusoid. (3 point).

   Solution: Defining the weight vector $w \in \mathbb{R}^{2K}$ and the feature vector $\Phi(t)$ as follows,

   $$w := (\alpha_1, \beta_1, \ldots, \alpha_K, \beta_K)^\top,$$
   $$\Phi(t) := (\cos(\omega_1 t), \sin(\omega_1 t), \ldots, \cos(\omega_K t), \sin(\omega_K t)^\top,$$

   we can rewrite our predictor $\hat{y}$ as a the following linear function:

   $$\hat{y} = w^\top \Phi(t) = \sum_{k=1}^{K} \alpha_k \cos(\omega_k t) + \beta_k \sin(\omega_k t).$$

3. Formulate the mean-square-error loss function. (2 point)

   Solution: The mean-square-error loss function is given by:

   $$L(w) = \frac{1}{T}\sum_{t=1}^{T}(w^\top \Phi(t) - y^t)^2.$$

4. Derive the gradient of the loss function with respect to the linear regression parameters α_k, β_k, $k = 1, 2, \ldots, K$. (4 point)

   Solution: The gradient is $\nabla L(w) = \left(\frac{\partial L(w)}{\partial \alpha_1}, \frac{\partial L(w)}{\partial \beta_1}, \ldots, \frac{\partial L(w)}{\partial \alpha_K}, \frac{\partial L(w)}{\partial \beta_K}\right)^\top$, where

   $$\frac{\partial L(w)}{\partial \alpha_k} = \frac{2}{T}\sum_{t=1}^{T}(w^\top \Phi(t) - y^t)\frac{\partial [w^\top \Phi(t)]}{\partial \alpha_k} = \frac{2}{T}\sum_{t=1}^{T}(w^\top \Phi(t) - y^t)\cos(\omega_k t),$$
   $$\frac{\partial L(w)}{\partial \beta_k} = \frac{2}{T}\sum_{t=1}^{T}(w^\top \Phi(t) - y^t)\frac{\partial [w^\top \Phi(t)]}{\partial \beta_k} = \frac{2}{T}\sum_{t=1}^{T}(w^\top \Phi(t) - y^t)\sin(\omega_k t).$$

5. Provide an approach to compute the optimal set of parameters without using gradient descent. (2 point)

Solution: Let us define the data matrix

$$\mathbf{\Phi} := \begin{pmatrix} \cos(\omega_1) & \sin(\omega_1) & \dots & \cos(\omega_K) & \sin(\omega_K) \\ \cos(2\omega_1) & \sin(2\omega_1) & \dots & \cos(2\omega_K) & \sin(2\omega_K) \\ \vdots & \vdots & & \vdots & \vdots \\ \cos(T\omega_1) & \sin(T\omega_1) & \dots & \cos(T\omega_K) & \sin(T\omega_K) \end{pmatrix},$$

and the observation vector $y := (y^1, ..., y^T)$. Since we are dealing with a linear regression problem, which is convex, we can use the Least Squares method to find the optimal parameters w^*:

$$w^* = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top y.$$

Problem 3.. (Dimensionality reduction using PCA [15 points])
Consider a dataset $\{x^i\}_{i=1}^N$, where $x^i \in \mathbb{R}^2$. *Assume the feature vectors are centered so each feature vector has zero mean.* Let σ_1^2, σ_2^2 denote the variance of features 1 and 2, respectively. Suppose that the second feature is a constant multiple of the first feature. So, for any data point $x^i \in \mathbb{R}^2$, we can write $x_2^i = \alpha x_1^i$, where $\alpha \in \mathbb{R}$ is an unknown constant, here assumed to be positive[1].

Here, we derive the visually obvious fact that the PCA projects the data onto the line $x_2 = cx_1$.

1. Derive the variance of $\{x_2^i\}_{i=1}^N$ denoted by σ_2^2, in terms of α and σ_1^2. (2 point)
   Using the variance calculation, show that the correlation between the two features is 1 irrespective of value of α. (2 point).

   Solution: We can use the following property of the variance to compute the answer right away:

   $$\sigma_2^2 = Var(x_2) = Var(\alpha x_1) = \alpha^2 Var(x_1) = \alpha^2 \sigma_1^2.$$

   Otherwise, we can see that since both features are centred, their means are $\mu_1 = \mu_2 = 0$. Starting from the definition of the variance, we can then write:

   $$\sigma_2^2 = \frac{1}{N}\sum_{i=1}^{N}(x_2^i - \mu_2)^2 = \frac{1}{N}\sum_{i=1}^{N}(\alpha x_1^i - \mu_1)^2 = \frac{1}{N}\sum_{i=1}^{N}(\alpha x_1^i - \alpha\mu_1)^2 = \alpha^2\frac{1}{N}\sum_{i=1}^{N}(x_1^i - \mu_1)^2 = \alpha^2\sigma_1^2.$$

   The correlation between the two features can then be computed as follows:

   $$corr(x_1, x_2) = \frac{Cov(x_1, x_2)}{\sqrt{\sigma_1^2}\sqrt{\sigma_2^2}} = \frac{Cov(x_1, \alpha x_1)}{\sqrt{\sigma_1^2}\sqrt{\alpha^2\sigma_1^2}} = \frac{\alpha Cov(x_1, x_1)}{\alpha\sqrt{\sigma_1^2}\sqrt{\sigma_1^2}} = \frac{Var(x_1)}{\sigma_1^2} = \frac{\sigma_1^2}{\sigma_1^2} = 1,$$

   where $Cov(x_1, x_2)$ denotes the covariance between the two features.

   If this formula is unknown, we can alternatively write:

   $$\begin{aligned} corr(x_1, x_2) &= \frac{Cov(x_1, x_2)}{\sigma_1\sigma_2} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_1^i - \mu_1)(x_2^i - \mu_2)}{\sigma_1\sigma_2} = \frac{\frac{1}{N}\sum_{i=1}^{N} x_1^i x_2^i}{\sigma_1\sigma_2} \\ &= \frac{\frac{1}{N}\sum_{i=1}^{N} x_1^i \alpha x_1^i}{\sigma_1\sigma_2} = \alpha\frac{\frac{1}{N}\sum_{i=1}^{N}(x_1^i)^2}{\sigma_1\sigma_2} = \alpha\frac{\frac{1}{N}\sum_{i=1}^{N}(x_1^i - \mu_1)^2}{\sigma_1\sigma_2} = \alpha\frac{\sigma_1^2}{\sigma_1\sigma_2}. \end{aligned}$$

   Given $\sigma_2^2 = \alpha^2\sigma_1^2$, this indeed gives:

   $$corr(x_1, x_2) = \alpha\frac{\sigma_1^2}{\sigma_1\sigma_2} = \alpha\frac{\sigma_1^2}{\alpha\sigma_1\sigma_1} = 1.$$

2. Derive the matrix $C = X^T X \in \mathbb{R}^{2\times 2}$ in terms of σ_1^2 and α. (1 point)

[1] In practice, due to noisy measurements, x_2 would only approximately be a constant multiple of x_1 in the dataset. Here, to simplify the computations we discard the impact of the noise and assume to have access to "perfect" data.

Solution: We have:

$$\begin{aligned} C = X^\top X &= \begin{pmatrix} x_1^1 & \dots & x_1^N \\ x_2^1 & \dots & x_2^N \end{pmatrix} \begin{pmatrix} x_1^1 & x_2^1 \\ \dots & \dots \\ x_1^N & x_2^N \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^N (x_1^i)^2 & \sum_{i=1}^N x_1^i x_2^i \\ \sum_{i=1}^N x_1^i x_2^i & \sum_{i=1}^N (x_2^i)^2 \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^N (x_1^i)^2 & \alpha \sum_{i=1}^N (x_1^i)^2 \\ \alpha \sum_{i=1}^N (x_1^i)^2 & \alpha^2 \sum_{i=1}^N (x_1^i)^2 \end{pmatrix} \\ &= \begin{pmatrix} N\frac{1}{N}\sum_{i=1}^N (x_1^i)^2 & \alpha N \frac{1}{N} \sum_{i=1}^N (x_1^i)^2 \\ \alpha N \frac{1}{N} \sum_{i=1}^N (x_1^i)^2 & \alpha^2 N \frac{1}{N} \sum_{i=1}^N (x_1^i)^2 \end{pmatrix} \\ &= \begin{pmatrix} N\sigma_1^2 & \alpha N \sigma_1^2 \\ \alpha N \sigma_1^2 & \alpha^2 N \sigma_1^2 \end{pmatrix}, \end{aligned}$$

where we used the fact that $\mu_1 = 0$ in the definition of σ_1^2.

3. Let $c_1, c_2 \in \mathbb{R}^2$ denote the columns of C. Verify that $c_2 = \alpha c_1$. (1 point)
Solution: Define c_1 as $c_1 = N\sigma_1^2(1, \alpha)^\top$. Then

$$C = (c_1, c_2) = \begin{pmatrix} N\sigma_1^2 & \alpha N \sigma_1^2 \\ \alpha N \sigma_1^2 & \alpha^2 N \sigma_1^2 \end{pmatrix} = N\sigma_1^2 \begin{pmatrix} 1 & \alpha \\ \alpha & \alpha^2 \end{pmatrix} = (c_1, \alpha c_1).$$

4. Using the result of Part 3, show that the vector $(1, \alpha) \in \mathbb{R}^2$ is an eigenvector of the matrix C and derive its corresponding eigenvalue. (2 point)
Solution: We have:

$$C \begin{pmatrix} 1 \\ \alpha \end{pmatrix} = (c_1, \alpha c_1) \begin{pmatrix} 1 \\ \alpha \end{pmatrix} = c_1 + \alpha^2 c_1 = (1 + \alpha^2) N\sigma_1^2 \begin{pmatrix} 1 \\ \alpha \end{pmatrix}.$$

Thus $C(1, \alpha)^\top = \lambda_1 (1, \alpha)^\top$, where $\lambda_1 = N\sigma_1^2(1 + \alpha)$ is the corresponding eignevalue to the eigenvector $(1, \alpha)^\top$.

5. Using the result of Part 3, show that the vector $(-\alpha, 1) \in \mathbb{R}^2$ is an eigenvector of the matrix C corresponding to an eigenvalue at 0. (1 point)
Solution: We have:

$$C \begin{pmatrix} -\alpha \\ 1 \end{pmatrix} = (c_1, \alpha c_1) \begin{pmatrix} -\alpha \\ 1 \end{pmatrix} = -\alpha c_1 + \alpha c_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = 0 \begin{pmatrix} -\alpha \\ 1 \end{pmatrix}.$$

Thus, $(-\alpha, 1)^\top$ is an eigenvector of C with corresponding eigenvalue $\lambda_2 = 0$.

6. Scale the first principal component of C to have a unit Euclidean norm. (1 point)
Solution: The first principal component of C is given by the eigenvector corresponing to the largest eigenvalue, i.e., $v_1 := (1, \alpha)^\top$. In the following the first principal component is scaled to unit euclidian norm:

$$\hat{v}_1 = \frac{v_1}{\|v_1\|} = \frac{1}{\sqrt{1 + \alpha^2}} \begin{pmatrix} 1 \\ \alpha \end{pmatrix}.$$

7. The projection of the data matrix $X \in \mathbb{R}^{N\times 2}$ onto its first principal component is given by $X\Theta \in \mathbb{R}^N$. Write what Θ is in this problem. (1 point)
Solution: Θ corresponds to the normalized first principal component, i.e., $\Theta = \hat{v}_1 = \frac{1}{\sqrt{1+\alpha^2}}(1,\alpha)^\top$.

8. Verify that the projection of each data point (x_1^i, x_2^i) onto the first principal component gives the Euclidean distance of (x_1^i, x_2^i) from the origin $(0,0)$ and draw a visualization of the result. (3 point).
Solution: On the one hand, the projection of each data point $(x_1^i, x_2^i)^\top$ onto the first principal component results in:

$$\begin{aligned}
\begin{pmatrix} x_1^i & x_2^i \end{pmatrix} \Theta &= \begin{pmatrix} x_1^i & x_2^i \end{pmatrix} \frac{1}{\sqrt{1+\alpha^2}} \begin{pmatrix} 1 \\ \alpha \end{pmatrix} \\
&= \begin{pmatrix} x_1^i & \alpha x_1^i \end{pmatrix} \frac{1}{\sqrt{1+\alpha^2}} \begin{pmatrix} 1 \\ \alpha \end{pmatrix} \\
&= \frac{1}{\sqrt{1+\alpha^2}}(1+\alpha^2)x_1^i \\
&= \sqrt{1+\alpha^2}x_1^i.
\end{aligned}$$

On the other hand, the Euclidian distance of $(x_1^i, x_2^i)^\top$ from the origin results in:

$$\|x^i - 0\| = \sqrt{(x_1^i - 0)^2 + (x_2^i - 0)^2} = \sqrt{(x_1^i)^2 + (\alpha x_1^i)^2} = \sqrt{1+\alpha^2}x_1^i,$$

which indeed confirms that the projection onto the first principal component corresponds to the distance from the origin.

9. What would look different in your projection if you first standardize the features to unit variance before PCA computations above? (1 point)
Solution: Since the mean of both features are $\mu_1 = \mu_2 = 0$ and using the result of point 1, we would get the standardized features for all $i = 1, \ldots, N$:

$$\begin{aligned}
\hat{x}_1^i &= \frac{x_1^i}{\sigma_1}, \\
\hat{x}_2^i &= \frac{x_2^i}{\sigma_2} = \frac{\alpha x_1^i}{\alpha\sigma_1} = \frac{x_1^i}{\sigma_1} = \hat{x}_1^i.
\end{aligned}$$

Since one feature is an exact multiple of the other, standardizing both reduces them to the same values! There is hence no need to do the PCA anymore, as the data is already "projected" in one dimension.

Problem 4. (Neural networks [13 points])

1. You want to construct a neural network that takes two binary inputs $x_1, x_2 \in \{0, 1\}$ and gives the output $y = 0$ if $x_1 = x_2 = 0$ and $y = 1$, otherwise. In other words, the neural network is implementing the logical "OR" function.

   Construct this network using a single hidden layer, with one node in the hidden layer and the unit step activation function, that is, $\sigma(z) = 1$ if $z \geq 0$ and $\sigma(z) = 0$, otherwise. Draw your network and specify its weight and bias values. (5 point)

   Solution: We can for example choose: $w_{1,1} = 1, w_{1,2} = 1, b_1 = -1$.

Now, we consider a robotic obstacle detection problem. You aim to control a robot car so that it can maneuver in a room while avoiding obstacles. Thus, you first want to design a neural network classifier that takes a camera image and labels objects identified as *obstacle* or *free space*. The car camera produces images of size 240×240 pixels over three $\{red, green, blue\}$ channels. For training your neural network, you take 200 images and manually label the objects in the image as *obstacle* or *free space*.

2. In your first attempt, you use a neural network with 4 hidden layers, where each hidden layer has 10 nodes and an output layer with 1 node. You use a *tanh* activation function and the logistic loss function for the output layer. You flatten each image before giving it to your neural network as an input. For the *first* layer of the network, how many weights and biases need to be determined? (2 point)

   Solution: The input dimension is $n_{\text{in}} = 3 \cdot 240 \cdot 240 = 3 \cdot 57,600 = 172,800$, and the first hidden layer has 10 nodes. Thus, there are $172,800 \cdot 10 = 1,728,000$ weights and 10 biases.

You find that the accuracy of your classifier above is not very good. So, you decide to use a pre-trained network and simply modify it for your task. You use the EfficientNet network, developed by Google, which is a convolutional neural network for classifying images into 1000 classes. The original network has 7.8 million parameters to train, and thankfully, the network has been trained already. The last layer of EfficientNet has 1000 outputs and the *softmax* activation function. You simply append one layer with a single output node to the last layer of EfficientNet, and you add no activation function to this node.

3. You use the logistic loss function for the last layer you added. Write the logistic loss function in terms of the output of the last layer of the EfficientNet and your newly added parameters. (2 point)

   Solution: Let's denote the output of the efficient net corresponding to the input x^i by $z^i \in \mathbb{R}^{1000}$. Then, the logistic loss is

   $$L(W, b) = \frac{1}{N} \sum_{i=1}^{N} y^i \log(1 + e^{-(Wz^i + b)}) + (1 - y^i) \log(1 + e^{Wz^i + b}),$$

   where W and b correspond to the weights and biases of the added layer.

4. Do you expect to be able to find the optimal parameters for the above problem using gradient descent? Justify your answer.s (1 point)

   Solution: Yes, because the problem is convex in W and b.

5. You observe the training loss oscillating throughout the iterations of the gradient descent. Explain why a decrease in the learning rate could result in a monotone decrease of the loss. (1 point)

   Solution: Oscillations typically occur when the learning rate is to high and we overshoot over the optimal solution. This problem can be avoided by reducing the learning rate.

6. Suppose that a filter of size 3×3 is used for convolution in the first layer and on the red channel of EfficientNet. The filter parameters are as shown in the matrix F below. Consider a 3×3 segment of the image over the channel *red* as shown in the matrix S below. What is the result of convolution of F with S? (2 point)

$$F = \begin{pmatrix} -1/4 & 0 & 1/4 \\ 0 & 0 & 0 \\ 1/4 & 0 & -1/4 \end{pmatrix}, \qquad S = \begin{pmatrix} 0.3 & 0.1 & 0.2 \\ 0.4 & .01 & 0.2 \\ 0.4 & 0.2 & 0.3 \end{pmatrix}$$

   Solution: The result of the convolution is:

$$F * S = -\frac{1}{4}\frac{3}{10} + \frac{1}{4}\frac{2}{10} + \frac{1}{4}\frac{4}{10} - \frac{1}{4}\frac{3}{10} = 0.$$

Problem 5. (Probabilistic classification [15 points])
At last, you understood decision trees and decided to make a youtube video to explain it to all those students who might be confused. Your video is posted and you are receiving many comments. You want to design an automatic way of analyzing the "sentiment" of the comments, that is, to understand if the watchers found your video helpful (positive +) or unhelpful (negative -).

So, you classify a few comments as below, and plan to let the machine learning algorithms do the analysis for the rest of the comments.

class label	comment
+	clear and engaging
-	confusing but funny
-	boring and many typos
+	engaging and funny
+	highly helpful and clear

Note: For the three points below, ignore the common word "and". Hence, the 5 comments above contain 10 distinct words.

1. Write the prior probability of positive and negative classes. (1 point)

   Solution:

$$\mathcal{P}(\text{Positive class}) = \frac{3}{5}$$
$$\mathcal{P}(\text{Negative class}) = \frac{2}{5}$$

2. Write the probability of the word "funny" given class positive and the probability of the word "funny" given class negative (2 point).

   Solution:

$$\mathcal{P}(\text{funny} \mid \text{Positive class}) = \frac{1}{3}$$
$$\mathcal{P}(\text{funny} \mid \text{Negative class}) = \frac{1}{2}.$$

3. Write the probability of "helpful and funny" given class positive using the Naive Bayes assumption. (1 points)

   Solution:

$$\mathcal{P}(\text{helpful and funny} \mid \text{Positive class}) = \mathcal{P}(\text{helpful} \mid \text{Positive class}) \times \mathcal{P}(\text{funny} \mid \text{Positive class})$$
$$= \frac{1}{3} \times \frac{1}{3} = \frac{1}{9}.$$

Now, suppose that you use 100 comments for training so that you encounter many more positive and negative comments and many more distinct words.

4. Your Naive Bayes approach classifies 3 of the positive comments as negative and 7 of the negative comments as positive. What is its error rate? (1 point)

   Solution: The error rate in that case is $\frac{3+7}{100} = \frac{10}{100} = 10\%$.

5. Your classifier has much more false positives than false negatives. Since you care more about the critical comments to improve your work, you decrease the *prior* probability of class positive. Show that this change could reduce the number of false positives. (2 point)

   Solution: Since we care more about critical comments, we want to detect more negative classes instead of positive classes. Therefore, we need reduce the number of false positives by decreasing $\mathcal{P}(\text{Positive class} \mid \text{word})$ which can be written as

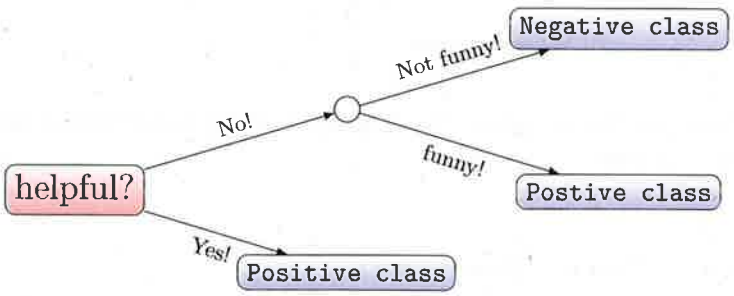
$$\mathcal{P}(\text{Positive class} \mid \text{word}) = \frac{\mathcal{P}(\text{word} \mid \text{Positive class})\mathcal{P}(\text{Positive class})}{\mathcal{P}(word)}.$$

   Therefore, decreasing the *prior* probability of class positive will reduce the number of false positives.

Suppose that you now replaced your Naive Bayes classifier with a decision tree, trained on the same 100 comments above and using the gini criterion. You found that the presence of the feature "helpful" had the lowest gini index. In particular, all the 20 comments that had the word "helpful" in them were of class positive. For the remaining 80 comments, you found that the word "funny" had the lowest gini index. In particular, 20 of the 80 comments did have the word "funny", and of these 20, 15 were of class positive; whereas 60 of the 80 comments did not have the word "funny" and of these 60, 20 were of class positive. Now, you stopped growing your tree, at depth 2.

6. Draw your decision tree. (2 point)

   Solution:



7. What was the gini index of splitting the "not helpful" node into a leaf node corresponding to "funny" and "not funny"? (2 point)

   Solution:

$$\text{Gini index} = \frac{20}{80}\left(\frac{5}{20}\frac{15}{20} + \frac{15}{20}\frac{5}{20}\right) + \frac{60}{80}\left(\frac{40}{60}\frac{20}{60} + \frac{20}{60}\frac{40}{60}\right) = \frac{123}{288}$$

8. What is the false positive and false negative rate of this depth 2 decision tree? (2 point)

   Solution:

Truth \ Prediction	Negative	Positive
Negative	40	5
Positve	20	20 + 15

$$\text{False positive} = \frac{5}{45} = \frac{1}{9},$$
$$\text{False negative} = \frac{20}{55} = \frac{4}{11}.$$

9. Comment on the advantages and disadvantages of the two classifiers above for this problem. (2 point)

   Remark: Note that this is open question, we will give positive points based on your reasonable answers.

   Solution: A decision tree is more interpretive compared to probabilistic classification. However, probabilistic classification has higher accuracy compared to decision tree in this case, with an error of 10% compared to 25% for the tree.