Problem 1. (Feature engineering, regression) [13 **points**]
A server is one of the main energy-consuming components of a data center. It has been found that the variables CPU denoted by $x_1 \in \mathbb{R}$, and the memory load denoted by $x_2 \in \mathbb{R}$, are two of the main contributing factors to the energy consumption of a server. You have made measurements of the CPU x_1^i, memory loads x_2^i, and energy consumption y^i, for $i = 1, 2, \ldots, 2000$ instances and aim to use linear regression to come up with a function that predicts a server's energy consumption. You randomly select 400 data samples for testing and the rest for training.

1. Standardize the features to zero mean and unit variance. (2 points)
   Solution: For each $j = 1, 2.$ given the mean $\mu_i = \frac{1}{1600}\sum_{i=1}^{1600} x_j^i$ and the standard deviation $\sigma_j = \sqrt{\frac{1}{1600}\sum_{i=1}^{1600}\left(x_j^i - \mu_j\right)^2}$, we can standardize the feature by using the following formula:
   $$\tilde{x}_j^i = \frac{1}{\sigma_j}\left(x_j^i - \mu_j\right),$$

2. You have found that increasing CPU and memory load have a multiplicative effect on energy consumption. Hence, you define a new feature : $\phi(x_1, x_2) = x_1 x_2$. Write the equation for a linear predictor in terms of the features $x_1, x_2, \phi(x_1, x_2)$. (1 point)
   Solution: The linear predictor is:
   $$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + b = \begin{bmatrix} b & w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix} = w^\top z,$$
   with $w, z \in \mathbb{R}^4$.

3. Write the regularized mean-square loss function for identifying the parameters of the model; use $\lambda \in \mathbb{R}$ for regularization. (1 point)
   Solution: The regularized mean-square loss function is:
   $$\begin{aligned} MSE(w) &= \frac{1}{1600}\sum_{i=1}^{1600}(y^i - w^\top z^i)^2 + \lambda w^\top w \\ &= \frac{1}{1600}(y_{train} - Z_{train} w)^\top (y_{train} - Z_{train} w) + \lambda w^\top w, \end{aligned}$$
   where $y_{train} \in \mathbb{R}^{1600}$ and $Z_{train} \in \mathbb{R}^{1600\times 4}$ are obtained by stacking the y^i and the $(z^i)^\top$ respectively.

4. Derive the gradient of the loss function with respect to the linear regression parameters. (2 points)
   Solution: The gradient of the mean-square loss function is:
   $$\frac{\partial MSE}{\partial w} = \frac{2}{1600} Z_{train}^\top (Z_{train} w - y_{train}) + 2\lambda w.$$

5. Write the equation characterizing the optimal parameters (2 points).

   Solution: To find the optimal parameters, we put the gradient of the mean-square loss function equal to zero and we solve with respect to w:

$$w = \left(1600\lambda I + Z_{train}^{\top} Z_{train}\right)^{-1} Z_{train}^{\top} y_{train}.$$

6. Write the approach for finding the optimal parameters using gradient descent. (1 point)

7. Which is likely to decrease the training error: increasing or decreasing λ and why? (1 point)

   Solution: To decrease the training error we need to decrease λ. We added the regularization parameter λ to reduce overfitting, so we use it to reduce the accuracy prediction on the training data and thus we increase the training error. By setting λ equal to zero we minimize the training error.

8. Assume we choose the optimal λ using 5-fold cross-validation. Let $\hat{y}^i$ denote the prediction and y^i denote the actual server energy consumption for a given data point. How would you compute the mean validation error over the 5-folds? (2 points)

   Solution: The mean validation error ϵ can be computed as the average of the error ϵ_f of each of the five folds

$$\epsilon = \frac{1}{5}\sum_{f=1}^{5} \epsilon_f = \frac{1}{5}\sum_{f=1}^{5}\sum_{i \in I_f} \frac{1}{N_f}(\hat{y}^i - y^i)^2,$$

   where I_f gather the indices of the validation points in fold f and N_f is the cardinality of that fold.

9. Based on the result of 5-fold cross validation on the 1600 data points in the training set shown below, for which λ is the test error more likely to be similar to the validation error? (1 point)

model	mean validation error	variance of error
λ_1	2.35	9.42
λ_2	1.30	4.16
λ_3	1.76	3.50

   Solution: It is λ_3, because it is the one with the lowest variance of error. 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.

Problem 2. (PCA, Neural network) [16 **points**]
You want to classify bird types based on their song (a motivation is that diverse bird species are important for health of a forest). You have collected a sample of $N = 400$ bird songs, where each song is a sequence of vibrations measured at 0.01 second intervals of time for a duration of 70 seconds. So, a data point is given by $s^i \in \mathbb{R}^{7000}$, $i = 1, 2, \ldots, 400$. Your environmentalist friend has labeled the bird type $\{A, B, C\}$ for each s^i.

As a first step, you apply a convolution on the signal (your idea is that convolution can act like a filter and capture certain frequency of the signal.)

1. Consider a segment of s^i, as $u^i = (11, 3, 5, 7, 2, 13, 17, 29, 23, 19) \in \mathbb{R}^{10}$ and the filter $h = (-1, 0, 1) \in \mathbb{R}^3$. Write the convolution of filter h with signal u^i, namely, $v^i = u^i \star h \in \mathbb{R}^{10}$ (Do zero-padding to the signal.) (2 points).

   Solution: To get an output in $\mathbb{R}^{10}$, we apply zero-padding to get $\tilde{u}^i \in \mathbb{R}^{12}$. The padded signal is $\tilde{u}^i = (0, 11, 3, 5, 7, 2, 13, 17, 29, 23, 19, 0) \in \mathbb{R}^{12}$. The convolution outcome is

$$c^i = (\tilde{u}^i \star h)_m = \sum_{n=0}^{2} \tilde{u}^i_{m+n}\, h_n = (3, -6, 4, -3, 6, 15, 16, 6, -10, -23) \in \mathbb{R}^{10}$$

   .

2. Let $s^j, s^{j'}$ be two songs, where s^j has higher variations (a bird type with high frequency tone) than $s^{j'}$. After applying h to s^j and $s^{j'}$, which signal is more likely to have larger magnitude? (1 point)

   Solution: The magnitude of the convolution result c^j tends to be greater than that of $c^{j'}$.

After the convolution above you obtain an extended signal $x^i \in \mathbb{R}^{14000}$, which is the stacked vector of an audio recording and its convolutions. Given the high dimension of the feature vector, you consider principal component analysis. Let $C = X^\top X \in \mathbb{R}^{14000 \times 14000}$ denote the data covariance matrix (assume you have already done data normalization).

3. How many eigenvalues does C have? What is the dimension of an eigenvector of C? (1 point)

   Solution: C has 14000 eigenvalues. The dimension of eigenvectors of C is 14000×1.

4. You find that 6 of the eigenvalues are much larger than the rest of the eigenvalues. How many principal components you would choose, and what would they each correspond to? (1 point)

   Solution: I would choose 6 principle components, which correspond to

   (a) the eigenvectors associated with the 6 largest eigenvalues.

   (b) the 6 directions with the greatest variance in the dataset.

5. Now, let Θ denote the projection matrix. (a) What is its dimension and what are its columns? (b) Write the step for finding the lower dimensional data matrix $A \in \mathbb{R}^{400 \times 6}$. (1 point)

   Solution: (a) The projection matrix $\Theta \in \mathbb{R}^{14000 \times 6}$, the columns are the eigenvectors above. (b) The lower dimensional matrix can be obtained with $A = X\Theta$.

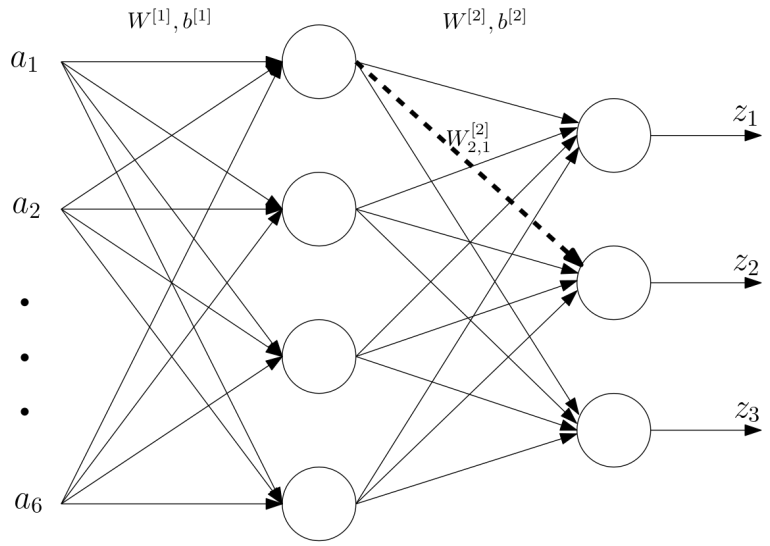
6. How would you evaluate the effectiveness of PCA above? (1 point)

   Solution. Look at reconstruction error $\|A\Theta^\top - X\|$.

Now, you consider a neural network (NN) to perform the classification given data matrix A. Consider a fully connected NN with 4 nodes and ReLU activation function in the first layer and 3 nodes in the second (output) layer with no activation function for this layer.

7. Draw the neural network (2 points). How many parameters need to be identified during the training of the network? (1 point)

   Solution:



   We need

   - First layer: $6 \times 4 = 24$ weights and 4 biases.
   - Second layer: $4 \times 3 = 12$ weights and 3 biases.

   In total, 43 parameters need to be identified during the training of the network.

8. Write the NN predictor by letting $W^{[j]}, b^{[j]}$ denote the weights and biases of the layers $j = 1, 2$. (2 points)

   Solution: Suppose $a \in \mathbb{R}^6$ is a reduced dimensional data. The NN predictor can be written as $f_w : \mathbb{R}^6 \to \mathbb{R}^3, \quad \text{with} \quad f_w(a) = {W^{[2]}}^\top \, g\left({W^{[1]}}^\top a + b^{[1]}\right) + b^{[2]}$, where $g(x) := \max(0, x)$ is the ReLU activation function.

9. Note that the output of the network is $z \in \mathbb{R}^3$. To perform classification, write the logistic loss (cross-entropy loss) function. *Hint:* for classification, we need to use the *softmax* function to convert the output to a probability distribution over the classes. (2 points)

   Solution: There are two different representations of the solution that use different notations of the class. Both solutions receive full marks.

(1) One-hot class encoding

First, let's encode the label of the data points using one-hot encoding:

- $y^i = \big[1, 0, 0\big]^\top$ represents the data point i belongs to class A.
- $y^i = \big[0, 1, 0\big]^\top$ represents the data point i belongs to class B.
- $y^i = \big[0, 0, 1\big]^\top$ represents the data point i belongs to class C.

Then, we use the softmax function to convert the NN output $z = \big[z_1, z_2, z_3\big]^\top \in \mathbb{R}^3$ to the predicted probability distribution over the 3 classes $\hat{y} = \big[\hat{y}_1, \hat{y}_2, \hat{y}_3\big]^\top \in \mathbb{R}^3$, where

$$\hat{y}_k = \text{softmax}(z_k) = \frac{e^{z_k}}{\sum_{l=1}^{3} e^{z_l}}.$$

The cross-entropy loss over the $N = 400$ data points is

$$L(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{3} y_k^i \cdot \log(\hat{y}_k^i)$$

(2) Enumeration class encoding

We use $y^i = 1, 2, 3$ to represent the data point i belongs to class A, B, and C respectively. An alternative representation of the cross-entropy loss is to use the indicator function as in the class note:

$$L(W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{3} \mathbf{1}\{y^i = k\} \cdot \log(\frac{e^{z_k}}{\sum_{l=1}^{3} e^{z_l}}).$$

10. Compute the gradient of the output of the last layer with respect to the weight connecting node 1 of the first layer to node 2 of the last layer, denoted by $W_{2,1}^{[2]}$. (2 points).

    Solution: Let's use subscript m to denote the m^{th} element of a vector. The gradient can be computed as

$$\frac{\partial z_2}{\partial W_{2,1}^{[2]}} = \frac{\partial}{\partial W_{2,1}^{[2]}} \left(\sum_{m=1}^{4} W_{2,m}^{[2]} \cdot \Big[g\left(W^{[1]\top} a + b^{[1]} \right) \Big]_m + b_2^{[2]} \right) = \Big[g\left(W^{[1]\top} a + b^{[1]} \right) \Big]_1 \in \mathbb{R},$$

    that is the output of node 1 of the first layer.

Problem 3 (Naive Bayes, k-Nearest Neighbor) [12 **points**]
Company X is hiring employees and aims to hire those who spend less time watching videos online. Thus, it wants to predict an applicant's potential to watch online videos based on past employee data. In particular, for each of the 1000 past employees, it has recorded whether they have a high GPA, and whether they watch online videos. Among those who do not watch videos, some play sports (none of those who watch videos play sports). The survey is summarized below.

	Employees	Sport	No Video
High GPA	600	150	300
low GPA	400	50	100

Let event A denote having a high GPA, event B denote playing sports, and event C denote not watching videos.

1. Calculate the following: (a) empirical probability of having a high GPA; (b) empirical probability of having a high GPA and playing sports. (1 point)

   Solution.

$$P(A) = \frac{600}{1000} = \frac{3}{5}$$
$$P(A \cap B) = \frac{150}{1000} = \frac{3}{20}$$

2. Show that the events A and B are not independent. (1 point)

   Solution.

$$P(A) = \frac{3}{5}$$
$$P(B) = \frac{150 + 50}{1000} = \frac{1}{5}$$
$$P(A \cap B) = \frac{3}{20}$$

   Thus

$$P(A)P(B) = \frac{3}{5} * \frac{1}{5} \neq \frac{3}{20} = P(A \cap B)$$

   which shows that events A and B are not independent.

3. Calculate the following: (a) the empirical conditional probability of event A given event C; (b) the empirical conditional probability of events A and B given event C. (2 points)

   Solution.
   (a)

$$P(A|C) = \frac{P(A \cap C)}{P(C)} = \frac{300}{400} = \frac{3}{4}.$$

(b)

$$P(A \cap B|C) = \frac{P(A \cap B \cap C)}{P(C)} = \frac{150}{400} = \frac{3}{8}.$$

4. Show that conditioned on C, the events A and B are independent. (1 point)

   Solution.

$$P(A|C) = \frac{3}{4}$$
$$P(B|C) = \frac{P(B \cap C)}{P(C)} = \frac{150 + 50}{400} = \frac{1}{2}$$
$$P(A \cap B|C) = \frac{3}{8}$$

   Thus $P(A \cap B|C) = P(A|C) * P(B|C) = \frac{3}{8}$ and conditioned on C, events A and B are independent.

The company aims to have a classifier for an employee by using the exact GPA ($x_1 \in \mathbb{R}_+$) and the average number of hours of sport played per week ($x_2 \in \mathbb{R}_+$). Based on past data, it fits two probability density functions conditioned on Y, where Y corresponds to watching videos ($Y = 1$) or not ($Y = 0$). These are denoted by $f_{x_1|Y} : \mathbb{R} \to \mathbb{R}$ and $f_{x_2|Y} : \mathbb{R} \to \mathbb{R}$.

5. Formulate the Naive Bayes classifier. (3 points)

   Solution. The probability that Y is 1 given x is given by

$$P(Y = 1|x) = \frac{f_{x_1|1}(x_1) * f_{x_2|1}(x_2) * P(Y = 1)}{f_x(x)}.$$

   Analogously, the probability that Y is 0 given x is given by

$$P(Y = 0|x) = \frac{f_{x_1|0}(x_1) * f_{x_2|0}(x_2) * P(Y = 0)}{f_x(x)}.$$

   Naive Bayes classifier classifies $Y = 1$ given x if

$$f_{x_1|1}(x_1) * f_{x_2|1}(x_2) * P(Y = 1) \geq f_{x_1|0}(x_1) * f_{x_2|0}(x_2) * P(Y = 0)$$

   and $Y = 0$ otherwise.

6. For an applicant, the company has obtained x_1, x_2 from which it has evaluated $f_{x_1|0}(x_1) = 0.25$, $f_{x_2|0}(x_2) = 2.00$, $f_{x_1|1}(x_1) = 0.20$ and $f_{x_2|1}(x_2) = 2.20$. Based on the Naive Bayes classifier, would this person be likely to watch online videos at work? (1 point)

   Solution. The probability that Y is 1 given x is given by

$$P(Y = 1|x) \propto f_{x_1|1}(x_1) * f_{x_2|1}(x_2) * P(Y = 1)$$
$$\propto 0.2 * 2.2 * \frac{600}{1000} \propto 0.264.$$

Analogously, the probability that Y is 0 given x is given by

$$P(Y=0|x) \propto f_{x_1|1}(x_1) * f_{x_2|1}(x_2) * P(Y=0)$$
$$\propto 0.25 * 2 * \frac{400}{1000} \propto 0.2.$$

Based on the Naive Bayes classifier this person is more likely to watch online videos when they start the job.

7. On a test set obtained from recently hired employees, it was found that the classifier has more false positives than false negatives. The company decided to change the prior on the probability of watching videos (perhaps the new generation has been bored of all the online videos). What term(s) in your Naive Bayes classifier would you change and how? (1 point)

   Solution. You could reduce the prior of class 1, namely decrease $P(Y=1)$ and in turn increase the prior on class 0, namely increase $P(Y=0)$. If the priors are updated based on new data and the conditional densities are calculated based on this data, then the conditional density functions $f_{x_1|Y}$ and $f_{x_2|Y}$ will change. However, since it is not specified how the conditional density functions $f_{x_1|Y}$ and $f_{x_2|Y}$ are computed in this problem we can not say whether they are affected or not by the change in the priors.

8. The error rate of the Naive Bayes classifier was found to be 0.80. The company aims to explore if the k-nearest neighbor approach could work better with $k=1$. For a person with $x_1 = 5.0$ and $x_2 = 3.0$, you are given two neighbors $x^i = (5.3, 3.0)$ with $Y^i = 1$ and $x^j = (5.2, 3.2)$ with $Y^j = 0$. Determine the label of the test point in the cases in which (a) the Euclidean distance; and (b) the Manhattan distance is used. (2 points)

   Solution.
   (a) Euclidian distance:

$$d(x, x^i) = \sqrt{(5.3-5)^2 + (3-3)^2} = 0.3$$
$$d(x, x^j) = \sqrt{(5.2-5)^2 + (3.2-3)^2} = \sqrt{0.08} < \sqrt{0.09} = 0.3$$

   $1-$nearest neighbor using the Euclidean distance labels the test point $x = (5, 3)$ as belonging to the same class as x^j, so to class 0.
   (b) Manhattan distance:

$$d(x, x^i) = |5.3-5| + |3-3| = 0.3$$
$$d(x, x^j) = |5.2-5| + |3.2-3| = 0.4 > 0.3$$

   $1-$nearest neighbor using the Manhattan distance labels the test point $x = (5, 3)$ as belonging to the same class as x^i, so to class 1.

Problem 4. (Decision-trees) [14 **points**]
Country X has observed that the number of people who vote in elections has been declining. The government decides to understand whether certain groups are less likely to vote and hopefully design schemes to improve this. To this end, it uses data of a population of 1000. For each individual, the data contains age x_1, education (no high school, high school, university) x_2, neighborhood (from 10 possibilities) x_3, income (low, medium, high) x_4, and voting y (yes or no).

1. Consider the problem of predicting whether an individual will vote based on the above features. What kind of machine learning problem is this? (1 point)

   Solution: This is a supervised classification.

2. For each of the features, state whether they are numerical, ordinal or categorical. (2 points)

   Solution: x_1 is numerical (we also accept ordinal), x_2, and x_4 are ordinal, whereas x_3 is categorical.

3. To design the decision tree, first consider the income. From the 600 people who have high income, 400 vote. What is the gini impurity of this leaf? (1 point)

   Solution: 400/600*200/600*2 = 4/9.

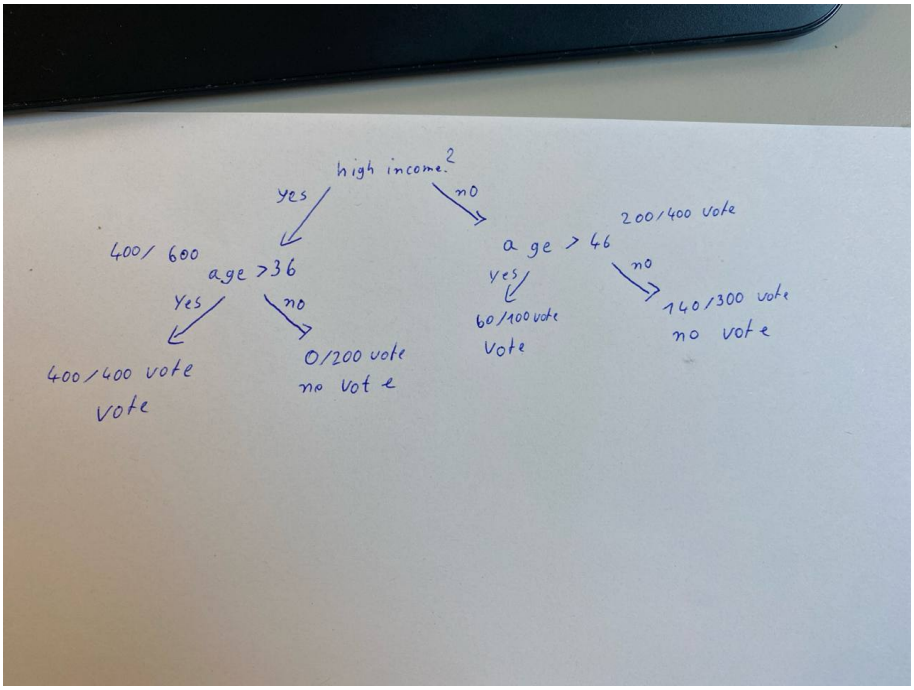
4. (a) For those who do not have high income, what should be the number of people who vote so that the gini impurity of the leaf will be 1/2? (.5 point) (b) What would be the resulting gini impurity of the node corresponding to "high income?"? (.5 point)

   Solution: (a) (200/400)*(200/400)*2=0.5, hence 200 people vote; (b) 6/10 * 4/9 + 4/10 * 1/2 =7/15

Let us fix the first node as "high income?" Among those with high income, all 400 aged above 36 vote. For those with not high income, 60 of 100 aged above 46 vote, whereas 140 of those aged below 46 vote.

5. Draw the decision tree with income as the first node and age based on thresholding as described above for creating the nodes at depth 2. Ensure to include the classification outcome at each leaf (2 points)

   Solution:,



6. We have a data from a person whose age is 25 and who has low income, what would the tree predict regarding whether this person would vote? (1 point)

   solution. no vote.

7. What is the accuracy of the above classifier? (1 point)

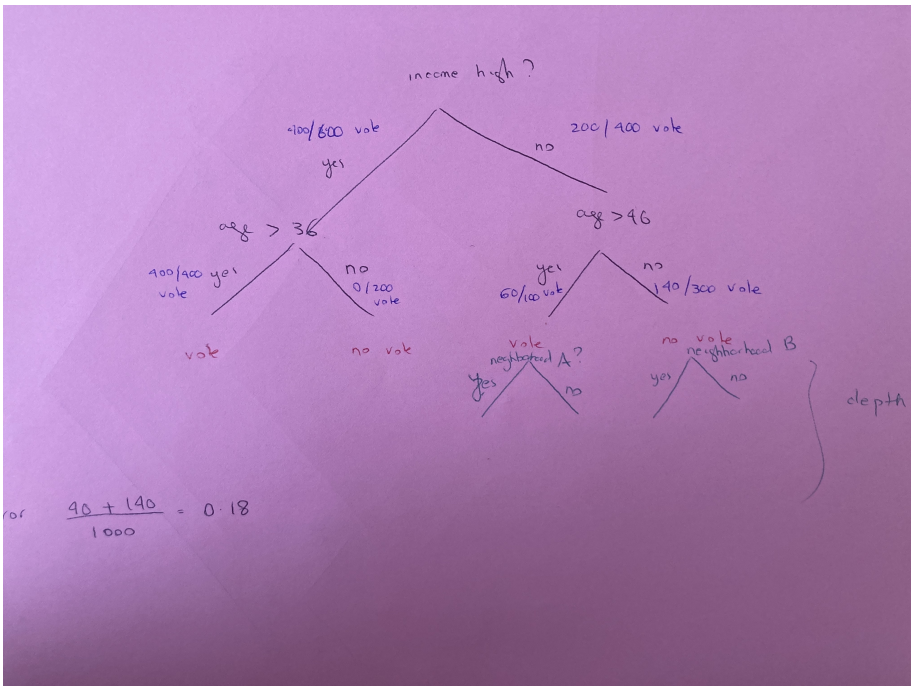
   solution. 1- (40 + 140)/1000 = .82

8. List 4 alternative approaches to decision trees that could be used to address this problem based on methods you have learned in this course. (2 points)

   Solution. logistic regression, k-NN, neural network, Naive Bayes

9. The government finds that a decision-tree is the most interpretable approach. It now wants to design a tree to choose one neighborhood in which it should strengthen outreach to the citizens about voting. Draw the tree with a single neighborhood being used as a feature in depth 3. (2 points)

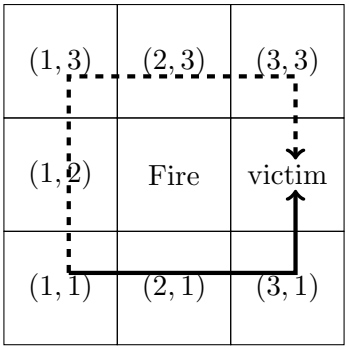
   Solution:, for all nodes that are not pure, we further split the data.



10. Consider now another decision tree that uses education and income as features, and a third one that uses education and age as features. Bootstrapping of the original data set is used to train each of these 3 trees. If you use the majority vote of the 3 trees for classification, do you expect the error rate to increase or decrease? (1 point)

    solution. should decrease because making an error requires 2 out of 3 classifiers make an error (see derivation in slides on the probability of this happening)

Problem 5. (Reinforcement Learning) [15 **points**]
There is a fire in a building, and a robot needs to plan its trajectory to rescue a victim. The building floor plan is shown in the figure below. The robot starts in state $s = (1, 1)$ with probability 1 and aims to rescue the victim at state $s = (3, 2)$. At each state s, the robot can choose from any feasible action $\mathcal{A}_s \subset \mathcal{A} = \{up, down, left, right\}$. So, for example, in $s = (1, 1)$, the robot can choose from $\mathcal{A}_{(1,1)} = \{up, right\}$. For an action taken at any state except $(2, 1)$, the robot will end up in the intended state with a probability 1, e.g. action *up* at state $s = (1, 2)$ leads to state $s' = (1, 3)$. For state $(2, 1)$, the robot ends up in the intended state with probability 70% and there is a 30% chance the robot might catch fire (as fire is spreading down), and the trajectory terminates. Once the robot reaches the victim, it gets a reward of 10 and the trajectory terminates. We consider a discount factor $0 < \gamma < 1$ (since with time, victim's health decreases). So, if the robot reaches the state $(3, 2)$ after t steps, she will get the reward of $\gamma^t \times 10$.



1. Write the transition probability $P(s'|s, right)$ (a) for $s = (1, 1)$; (b) for $s = (2, 1)$. (2 points)

   Solution: $P((2, 1)|(1, 1), right) = 1$, (consequently, $P(s'|(1, 1), up) = 0$ for any $s' \neq (2, 1)$). On the other hand, $P((3, 1)|(2, 1), right) = 0.7$, $P((2, 2)|(2, 1), right) = 0.3$.

2. For the solid trajectory τ_l, and the dashed trajectory τ_d, write the reward. Which trajectory has a higher reward? (2 points)

   Solution: Rewards for these two trajectories are

   $$R(\tau_d) = 10\gamma^5,\ R(\tau_l) = 10\gamma^3.$$

   Therefore, the solid trajectory has higher reward since $0 < \gamma < 1$.

3. Consider the sequence of deterministic actions leading to the trajectories τ_l and τ_d above. For trajectory τ_d this sequence is $u_d =$ (*up, up, right, right, down*). What is the sequence u_l leading to trajectory τ_l? (1 point)

   Solution: The sequence of actions for trajectory τ_l is (*right, right, up*)

4. Write the probabilities of getting the trajectories τ_l and τ_d given the sequence of actions u_l, u_d, respectively. (3 point).
   Hint: For a sequence of actions $u = (a_0, \ldots, a_{H-1})$ the probability of choosing the trajectory $\tau := (s_0, a_0, s_1, a_1, \ldots, s_H)$ is $\Pr(\tau) = \prod_{t=1}^{H} P(s_t|s_{t-1}, a_{t-1})$.

Solution:

$$\Pr(\tau_l) = 0.7,\ \Pr(\tau_d) = 1.$$

5. Write the expected reward for the sequence of actions u_l and u_d. (2 points).

   Solution:

$$\mathbf{E}(R_{u_d}) = R_{\tau_d} \times \Pr(\tau_d) = 10\gamma^5,$$
$$\mathbf{E}(R_{u_l}) = R_{\tau_l} \times \Pr(\tau_l) = 7\gamma^3.$$

6. How large should γ be such that the expected reward of u_d, leading to the longer but safer trajectory, becomes greater than the expected reward of u_l? (1 point)

   Solution: If $\gamma > \sqrt{\frac{7}{10}}$, we have $\mathbf{E}(R_{\tau_d}) > \mathbf{E}(R_{\tau_l})$. Therefore, the expected reward of trajectory τ_d becomes larger than the expected reward of trajectory τ_l if γ is large.

Now, we lay the steps towards learning the optimal softmax policy using policy gradient.

7. Write the softmax policy for $s = (1, 1)$ and $a = up$. (1 point)

   Solution:

$$\pi_\theta(up|(1,1)) = \frac{\exp\left(\theta_{(1,1),up}\right)}{\sum_{a' \in \mathcal{A}_{(1,1)}} \exp\left(\theta_{(1,1),a'}\right)} = \frac{\exp\left(\theta_{(1,1),up}\right)}{\exp\left(\theta_{(1,1),up}\right) + \exp\left(\theta_{(1,1),right}\right)}.$$

8. Note that $\theta \in \mathbb{R}^{14}$ as there are 2 possible actions at every state. Set θ to $\mathbf{0} \in \mathbb{R}^{14}$. Compute the stochastic policy gradient for the state $s = (1, 1)$ and action up with $\alpha = 1$ based on the trajectories τ_l, τ_d above (3 points).

   Solution: The formula for the component $(1, 1), up$ of the stochastic policy gradient is

$$\left(\hat{\nabla}_\theta J(\pi_\theta)\right)_{(1,1),up} = \frac{1}{2} \sum_{i \in \{d,l\}} R(\tau_i) \left(\sum_{t=0}^{H_i} \frac{\partial \log \pi_\theta(a_t^i | s_t^i)}{\partial \theta_{(1,1),up}} \right). \tag{0.1}$$

Furthermore, the derivative of the log policy is

$$
\begin{aligned}
&\frac{\partial \log \pi_\theta(a|s)}{\partial \theta_{(1,1),up}} \\
=&\frac{\partial}{\partial \theta_{(1,1),up}} \left(\log \frac{\exp\left(\theta_{s,a}\right)}{\sum_{a' \in \mathcal{A}_s} \exp\left(\theta_{s,a'}\right)} \right) && \text{(by the definition of the softmax policy parameterization)} \\
=&\frac{\partial}{\partial \theta_{(1,1),up}} \left(\theta_{s,a} - \log \sum_{a' \in \mathcal{A}_s} \exp\left(\theta_{s,a'}\right) \right) && \text{(by the additive property of the logarithmic function)} \\
=&\mathbf{1}_{(s,a)=((1,1),up)} - \mathbf{1}_{s=(1,1)} \frac{\exp\left(\theta_{s,up}\right)}{\sum_{a' \in \mathcal{A}_s} \exp\left(\theta_{s,a'}\right)} && \text{(basic derivative rules)} \\
=&\mathbf{1}_{(s,a)=((1,1),up)} - \mathbf{1}_{s=(1,1)} \pi_\theta(up|s) && \text{(by the definition of the softmax policy parameterization)} \\
=&\mathbf{1}_{s=(1,1)} \left(\mathbf{1}_{a=up} - \pi_\theta(up|s) \right). && \text{(take } \mathbf{1}_{s=(1,1)} \text{ outside of the equation)} \quad (0.2)
\end{aligned}
$$

Using above equation, we have

$$
\frac{\partial \log \pi_\theta(a|s)}{\partial \theta_{(1,1),up}} = \begin{cases} 0.5 & s = (1,1), a = up \\ -0.5 & s = (1,1), a = right \\ 0 & \text{otherwise.} \end{cases}
$$

Here, $\mathbf{1}_{\mathcal{A}}$ denotes an indicator function, which equals 1 when $\mathcal{A}$ is true and 0 otherwise. Hence, plugging this back into the formula (0.1) we arrive at

$$
\left(\hat{\nabla}_\theta J(\pi_\theta) \right)_{(1,1),up} = \frac{1}{2} \left(10\gamma^5 \cdot 0.5 + 10\gamma^3 \cdot (-0.5) \right).
$$