PY-599 (Fall 2018): Applied Artificial Intelligence
Midterm Exam

Closed books, closed notes, closed electronic-devices, and unfortunately closed neighbor!

NAME:
Department:

1- Which of the items below is an AI (machine learning) problem? Please place a check mark (√) in the answer boxes that correspond to your responses (There can be more than one correct option. Please choose all correct options). If you have any unconventional, controversial opinion, you should explain your answer, "*justifications*" after grading will not be accepted! (8 points)

☐ You have an online store, and you have your shoppers' purchase patterns with their profile (age, gender, zip code, etc.). A new shopper signs up and starts to search for items to purchase. The problem you like to solve is that based on shoppers' age, gender, zip code, etc. how to predict which items this new shopper would like better so that you put those items first in the search results.

☐ We are dropping a ball from a height to the ground. We know the ball's mass, the acceleration due to gravity (g), the air resistance force, height, and any other physical parameter needed. We would like to know at which velocity the ball would hit the ground?

☐ We like to design a program that monitors the images received from a camera attached to the front of a car and detects whether the items in front of the car are a) car, b) pedestrian, c) bicycle, or d) pole/tree.

☐ We like to design an image processing program that can automatically detect a tumor in an MRI image. We have access to a pile of MRI images with tumors and without tumors.

2- In (supervised) machine learning we usually split the dataset to three sets:

- Training Data
- Development Data (also known as Verification Data)
- Test Data

Please explain why we do that and how we use these three sets of data in practice? (15 points)

3. In (supervised) machine learning we usually deal with two types of problems: Regression Problems, and Classification Problems. *Define them* briefly in your own words. (10 points)

4. Which of the items below is a regression problem? Please place a check mark (√) in the answer boxes that correspond to your responses (Please leave classification problems unchecked). (8 points)

□ Predicting the price of a house based on its size, number of bedrooms, zip code, and other features.

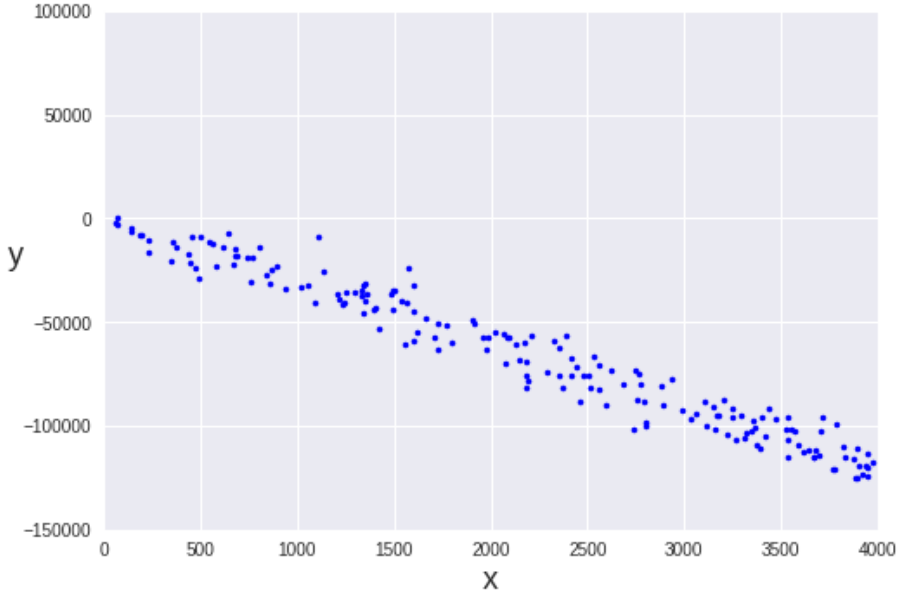
□ Assigning each image of a handwritten digit to a category (0, 1, 3, …, 9).

□ Determining whether a review is positive or negative.

□ Forecasting the temperature value based on historical data.

5. We had not officially defined the term ***supervised*** machine learning in the class. Basically, in supervising learning the training data that you feed into the algorithm includes the desired outputs (or labels) as well. For example, in sentiment analysis homework, where you developed a Naïve Bayes Classifier, each training example came with its correct labels: Positive or Negative. In ***unsupervised*** learning, the data is unlabeled. And the learner tries to learn from this unlabeled dataset. So far in the class all of our examples and homeworks were about the supervised machine learning techniques even though we never used the adjective "*supervised"* explicitly.
Right now, as your latest homework you are working on MNIST data set and trying to design a deep, feedforward neural network as a classifier. Is this a supervised machine learning or an unsupervised machine learning technique? Please explain yourself. (5 points)

6. Imagine you are given a regression problem, where x is the input, and y is the output you like to predict. You develop your python program that is composed of a linear regression model with a gradient descent optimization method to iteratively adjust and fit the parameters of the model to the training data. You print the cost function value at each iteration, and you observe that the cost function goes lower and lower at each iteration, and at some point, it reaches to some value and stays there (barely moves). This suggest that your model has hit a minimum point where gradient is zero, and hopefully the parameter values at this minimum point are the optimal parameters for your model to solve this regression problem.
In order to observe and visualize the performance of the system, and assess the quality of the solution, you plot the training data (y versus x) as is shown in the figure below, and then you plot the outputs of your linear regression model $\hat{y}$ for different x values. Ideally, what do you expect the outputs of your linear regression model ($\hat{y}$) for different x values look like? Please plot it over the training data in the figure below. (7 points)



7. Question 7 is the same as question 6, but with this difference that this time it is about classification. We like to see ideally what line our discriminative classifier should plot to separate two classes from each other. More details are below:

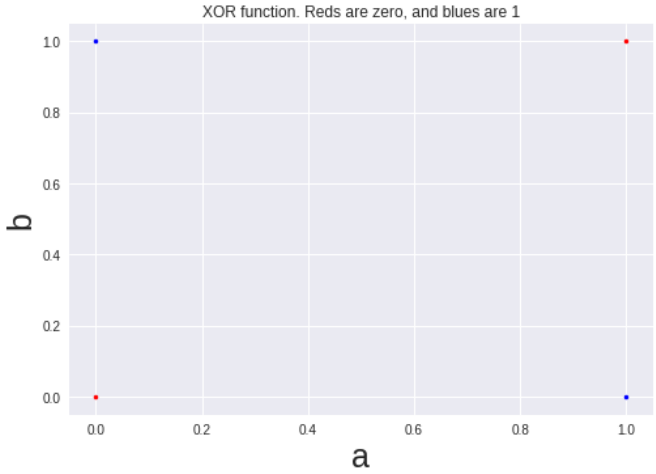
Imagine you are given a classification problem, where each data point has two features x_1 and x_2, and each data point belongs to one of two possible classes, class 0 or class 1. You develop your python program that is composed of a perceptron model (which is a linear model) with a step function as an output function, and a gradient descent optimization method to iteratively adjust and fit the parameters of the model to the training data. This is pretty much similar to your second homework, where you had a perceptron. You print the cost function value at each iteration, and you observe that the cost function goes lower and lower at each iteration, and at some point, it reaches to some value and stays there (barely moves). This suggest that your model has hit a minimum point where gradient is zero, and hopefully the parameter values at this minimum point are the optimal parameters for your model to solve this regression problem.

In order to observe and visualize the performance of the classifier, and assess the quality of the solution, you plot the training data as is shown in the figure below, where class 0 is in red and class 1 is in blue. The perceptron model in 2-D feature space is basically a line; it classifies the data points above the line as class 1, and the points below goes to class 0. Ideally, what do you expect this line to be? Please plot the ideal line of your trained linear model to classify the training data in the figure below. (7 points)

8. XOR function is a binary function that implements the following truth table:

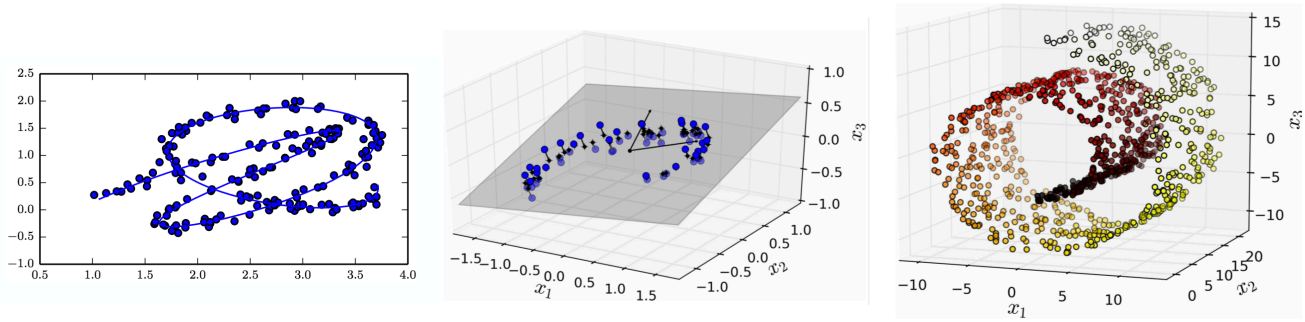
a	b	c
0	0	0
0	1	1
1	0	1
1	1	0

The figure below shows the outputs of XOR function for different input sets. Blue is output 1, and red is output 0:

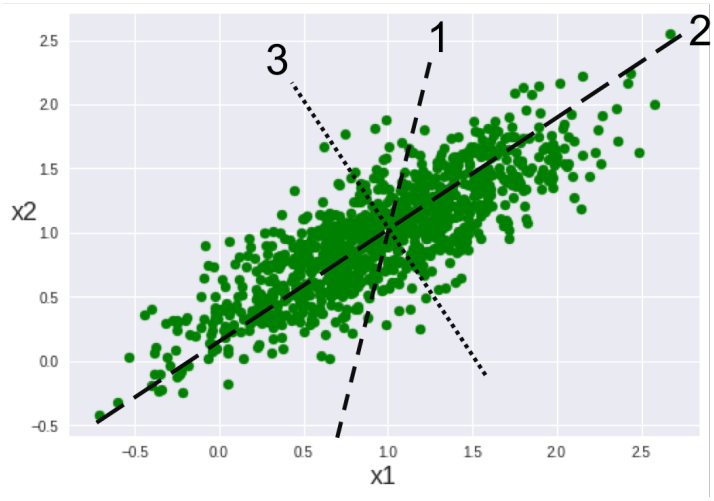


Do you think a linear classifier such as logistic regression can implement XOR function? Please explain your answer. What is the maximum accuracy (the ratio of correct output divided by total) that a linear classifier can achieve when it tries to model XOR function? (10 points)

9. Manifold Assumption says that the real-world high-dimensional datasets that come from an experiment or generated from a system do not randomly spread all over the place, instead there is some order or relationship between different data points in this dataset and therefore the dataset lies close to a much lower-dimensional manifold. A few examples are shown below.



This suggests that we can potentially project the dataset from the high dimensional state space to this lower-dimensional state space, where we have to deal with fewer features and therefore processing would be easier. There are many different dimension reduction techniques to perform this task. In the class we discussed PCA. In the figure below we observe a dataset, where each datapoint has two features, x_1 and x_2, and we have three possible axes to project the dataset into. In your opinion, which axis should we choose and why? (10 points)



10. In machine learning there is a fundamental tension between optimization and statistical generalization. Optimization refers to the process of adjusting model parameters in order to minimize the cost function (the error function) over the training data, whereas statistical generalization, or simply generalization, refers to how well the trained model performs over the new unseen data. We can design and train a model that performs perfectly on the training data, but extremely poorly over the test data. Simply put such a model is completely useless! In such cases, the model goes beyond trying to learn the general patterns of the data, and starts to learn the patterns that are specific to the training data–the noise. This phenomenon is called overfitting.

So far in the class we have had a very reactive interaction with statistical generalization; we ran the optimization techniques with no attention to generalization and just tried to minimize the cost function, and then we tried the trained model on the test data to assess the performance and to see whether it generalizes well on the new unseen data or not. There are more proactive methods to insure generalization. In these approaches we consider generalization as a factor during optimization. Such methods to insure generalization are called regularization methods. One of the main ideas behind regularization is that, if a learning model can only afford to learn and memorize a small number of patterns in the data, the optimization process will force it to focus more on the more general, prominent patterns instead of small, training data specific patterns and noise. This method would result in better generalization.

Loosely speaking, the higher the number of parameters of a model, the more powerful the model is and it has a higher learning capacity, and it is more likely to overfit to the training data. One the other side, if a model has very few parameters, it may not manage to learn even the general patterns of the data. We call such cases underfitting. As a result, we are looking for a model that has enough learning capacity to learn the main patterns and reduce the error function over the training data, but not too many parameters and too much learning capacity to overfit. So, what if we add a new term to the cost function to incorporate these two conflicting factors in one place and try to find a good tradeoff solution. For example, consider the new cost function for a regression problem:

$$J(W) = \frac{1}{2m}\sum_{i=1}^{m}\left(x_p^i.W - y^i\right) + \alpha\|W\|^1$$

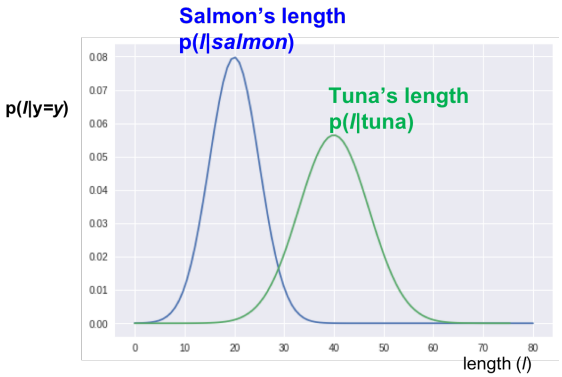
The right-hand side of the new cost function is composed of two parts, first the summation of error over the training data. We are very familiar with this cost function. By minimizing this part, we used to ensure that the model fits the training data. But now we have a new term as well, the L1 norm of the parameter vector. A norm is a function that assigns a nonnegative scalar value to a vector as a measure of its size. A zero vector – a vector with all zero elements– has a size of zero. As the absolute values of the elements of a vector grow, the norm of the vector grows as well. Long story short, by including a norm of the parameter vector (here L1 norm) to the cost function, we are penalizing models that have many large, non-zero parameters. A model with many non-zero parameters can overfit the training data. we do not want that! So we are forcing unnecessary parameters to become zero. On the other hand, if too many parameters of a model are zero, meaning that the model has very few active parameters, then the model may not fit the training data well enough, and as a result the first term, $\frac{1}{2m}\sum_{i=1}^{m}\left(x_p^i.W - y^i\right)$, is going to be high. By combining these two competing factors in one cost function, we force the optimizer to find us a sweet spot between these two extreme cases, a model that neither overfits nor underfits the training data, instead it just fits the training data. α is a nonnegative hyperparameter. Instead of L1 norm other norms could be used too. In session 14 we talked about Lasso model. Lasso is linear regression model with L1 norm for regularization, which is basically what we just explained here.

Do you think this idea can help us to improve statistical generalization of machine learning models? Please place a check mark (√) in the answer box that correspond to your response (10 points).

☐ Yes

☐ Si

11. Suppose a fish-packing plant is trying to automate the process of sorting incoming fishes on a belt according to species. And this plant receives just two types of fish, salmon and tuna. And they hire you to implement the classifier part. You go to the plant, gather hundreds of fishes from both categories, and start to study their feature in order to come up with informative, discriminative features that can help you to classify them to salmon and tuna.
You realize that the tuna fish tends to be longer than the salmon. And it can be a good feature to measure and use in your classifier. You estimate the class-conditional probability density functions for each class of fish from your pool of training fishes (which is basically normalized histograms of length of fishes for each class of fish) as you can see below.



Also, you are told that at this time of the year the probability of catching a tuna is 0.1 (P(tuna)=0.1), whereas the probability of catching a salmon is 0.9 (P(salmon)=0.9). This is your prior knowledge! You install some automatic length measuring instruments on the belt so that whenever a fish comes in, you get a reading on the length of the fish.
Part 1:
Now assume a new fish shows up on the belt. And your system needs to decide which class it belongs to, salmon or tuna. Assume there is a problem in your length measuring instrument and you are unable to measure the length of the fish and you must make a decision immediately before the fish goes through. So without knowing anything about the length or any other feature of the fish, how should you classify this fish? And what would be your error rate? Assume the only thing that you care is to keep the number of misclassifications as low as you can. (5 points)

Part 2:
Now assume your length-measuring instrument is up and running again and it tells you the length of the fish is 30 inches. Your class conditional probability density functions (the figure above) tells you that P(30 inches| slamon)=0.01 and P(30 inches| tuna)=0.02. So now with this additional information that we learned and knowing the prior information that we had about P(salmon) and P(tuna), how should we classify this fish? (5 points)