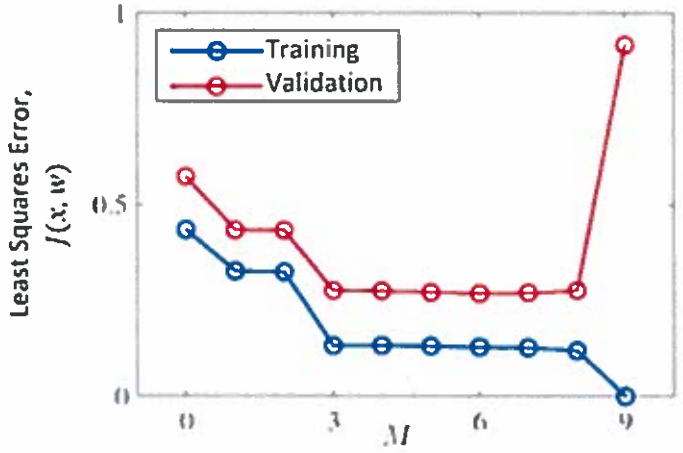
# Practice Problems

## Problem 1 – Polynomial Regression

Consider the polynomial curve fitting example discussed in class:

$$y(x, w) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=1}^{M} w_j x^j$$

Suppose you have a data set, $X$, that you split into two groups once: a fixed training subset, $X$_train, and a fixed validation subset $X$_validation. After fitting the polynomial to the training set across a range of model orders and evaluating on both training and validation sets, you obtain the following plot.




Answers the following questions:

1. Based on this plot, provide a discussion about which model order, $M$, should be used to avoid overfitting.
2. Since you have split your data into training and validation sets, are you safe from overfitting or can you still overfit? Explain your reasoning.
3. What are the common approaches to avoid overfitting given a model?
4. For the data set $X = \{x_1, x_2, \ldots, x_N\}$ and the labels $t = \{t_1, t_2, \ldots, t_N\}$, what objective function can you use if you want to optimize the polynomial linear regression model with a regularizer on the weights $w = [w_1, w_2, \ldots, w_M]^T$? Write down the equation for this objective function.

① We want to choose a model order that minimizes the ERROR in training and is still able to generalize to new da

So, we will choose the model order, M, that minimizes both training and validation ERROR. With this, we have the options $M = \{3, 4, 5, 6, 7, 8\}$, all with the same ERROR value.

Because we can still overfit the model, we will choose

the simplest model that allows for generalization.

Therefore, we choose $M = 3$.

(2) We can still overfit. Specifically if we are dealing with small data set or noisy data (as the model tends to learn all noise).

Cases where the test set is sampled from a region outside the input space region will cause issues in model prediction and/or data characterization.

(3) The common approaches to avoid overfitting are:

a) Cross-validation.

Split the training set into training and validation sets to evaluate generalization ability as model parameters are changed.

b) Regularization.

Add a penalty term on the model parameters.

c) Collect more data.

(4)

$$J(x, w) = \frac{1}{N} \sum_{i=1}^{N} \left( \sum_{j=0}^{M} w_j \cdot \phi(x_j) - t_i \right)^2 + \lambda \cdot \sum_{j=0}^{M} w_j^2$$

↳ Mean-squared error objective function with L2-norm penalty.

## Problem 2 – Curse of Dimensionality

**What is the curse of dimensionality? Why does it cause problems for certain Machine Learning algorithms? Describe two approaches to address curse of dimensionality. In your description state why each method is effective and what are its limitations.**

The Curse of Dimensionality refers to various phenomena that arises when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings. In ML, the common theme of these problems is that when the dimensionality increases, the volume of the space increases so fast that the available data becomes sparse. This sparsity is problematic for any method that requires statistical significance. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially the dimensionality.

Common approaches to address the curse of dimensionality are:

a) Collect more data
b) Decorrelate feature space (PCA)
c) Dimensionality Reduction (PCA, LDA).
d) Feature Selection (L1-norm, SBS)
e) Regularization (L1-norm, L2-norm, ...)

## Problem 3 – Curse of Dimensionality

Recall our discussion of the volume of the crust, i.e., the case of a sphere $S_2$ of radius $r - \epsilon$ inscribed within another sphere $S_1$ of radius $r$ and the relative volume of the crust and the outer sphere as we increase dimensionality $D$:

$$\frac{V_{\text{crust}}}{V_{S_1}} = \frac{V_{S_1} - V_{S_2}}{V_{S_1}} = 1 - \left(1 - \frac{\epsilon}{r}\right)^D$$

Describe (in paragraph form, be clear and thorough) how this concept relates to number of data points needed during classification as we increase the number of features we use for classification. In your discussion, be sure to answer: (1) the case when feature are uncorrelated, (2) the case when features are strongly correlated (e.g., $f_1 = af_2 + bf_3$), and (3) why this is an important issue in machine learning in general.

For a fixed radius R and parameter $\epsilon$, and $R \gg \epsilon$:

$$\lim_{D \to \infty} \frac{V_{\text{CRUST}}}{V_{S1}} = \lim_{D \to \infty} 1 - \left(1 - \frac{\epsilon}{R}\right)^D = 1$$

This indicates that as the dimensionality increases, all the volume will be in the corners of the space. So we will need exponentially more data to explain the entire space.

If the features are correlated, then we can apply compression / dimensionality reduction techniques without loosing significant information.

If the features are uncorrelated, then performing dimensionality reduction may result in information loss.

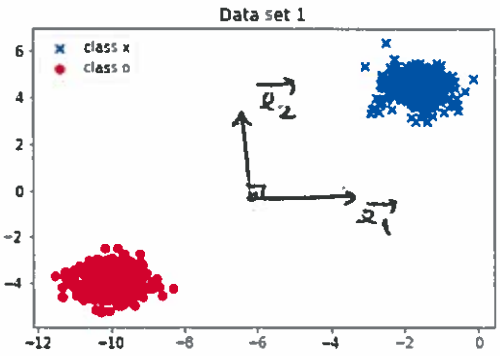
The Curse of dimensionality is an important issue in ML as it relates the number of samples with number of features and overfitting.

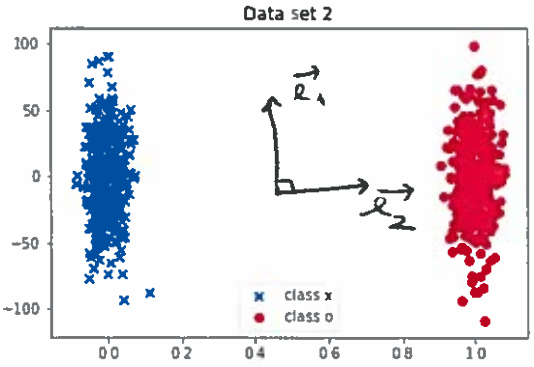
## Problem 4 – Principal Component Analysis

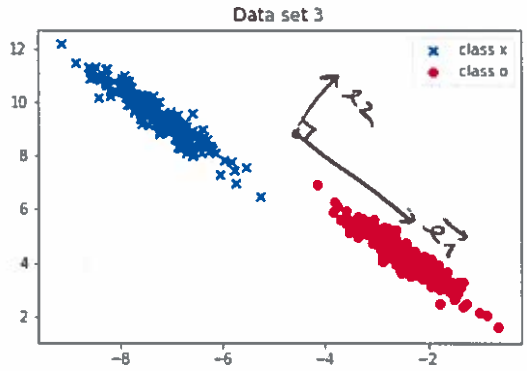
Consider the following four two-dimensional data sets each containing two clusters of data points (shown with "circles" and "crosses"). Suppose you would like to apply Principal Component Analysis (PCA) to reduce the dimensionality of each of these data sets from 2-D to 1-D where the two clusters remain separated in the 1-D projection. For each data set, address each of the following questions:

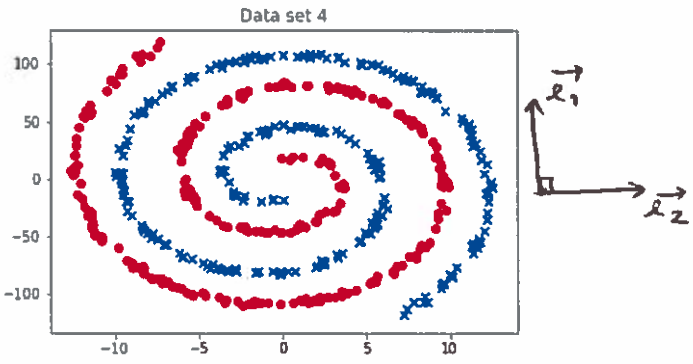
1. Draw the eigenvectors on each figure and clearly identify the (axis) direction of 1-D PCA projection.
2. Will PCA be effective at keeping the two clusters separated in the 1-D projection? Why or why not? If yes, state what characteristics of the data set allow PCA to be effective. If no, state what characteristics of the data set cause PCA to fail.
3. Can you think of another technique that would be successful at reducing the dimensionality of this data set while maintaining (or increase) separation between the two clusters? State the other method and describe why it would be successful.

(1)




$\vec{e}_1$ and $\vec{e}_2$ are the eigenvectors of the covariance matrix of the data $X$.

$\vec{e}_1$ is the direction of the 1-D PCA projection (1st principal component).

(2) DATA SET 1: YES. The clusters are perfectly separated and compact.

DATA SET 2: No. The direction of maximal variance ($\vec{e_1}$) will not be effective at keeping the clusters separated.

DATA SET 3: Yes. The direction of maximal variance ($\vec{e_1}$) aligns with the direction of maximum separability

DATA SET 4: No. The data classes have a non-linear relationship. PCA won't be able to preserve that.

(3) For data set 2, Linear Discriminant Analysis (LDA) will be able to project the data onto the direction of maximum class separability.

For data set 4, we can transform the features using some kernel function, say $\phi(x) = \sin(x)$, and then apply a simple LDA.

## Problem 5 – Principal Component Analysis

Suppose $X$ is a zero-mean data set of size $2 \times N$ with covariance matrix $R_X = \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix}$.

1. Write down the eigenvalue/eigenvector equation.
2. Calculate the eigenvalues $\lambda_1$ and $\lambda_2$ associated with the eigenvectors $u_1 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$ and $u_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, respectively, of $R_X$.
3. Suppose $Y$ is the PCA transformation of $X$. Write the formula for computing $Y$ from $X$. Be as specific as possible, you may use information from earlier parts.
4. What is the covariance matrix of $Y$? Why? Be as specific as possible, you may use information from earlier parts.
5. What is the amount of explained variance of the 1-D PCA projection?

(1) $R_x \cdot \vec{v} = \lambda \cdot \vec{v}$ , where $R_x$ : covariance of $X$
$\lambda$ : eigenvalue of $R_x$
$\vec{v}$ : associated eigenvector of $R$

(2) $R_x \cdot \vec{u_1} = \lambda_1 \cdot \vec{u_1} \Leftrightarrow \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \lambda_1 \begin{bmatrix} 2 \\ -1 \end{bmatrix} \Leftrightarrow \begin{cases} 2 = \lambda_1 \cdot 2 \\ 2 - 3 = -\lambda_1 \end{cases}$

$\Rightarrow \boxed{\lambda_1 = 1}$

$R_x \cdot \vec{u_2} = \lambda_2 \cdot \vec{u_2} \Leftrightarrow \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \lambda_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Leftrightarrow \begin{cases} 0 = \lambda_2 \cdot 0 \\ 3 = \lambda_2 \end{cases}$

$\Rightarrow \boxed{\lambda_2 = 3}$

(3) $Y = A \cdot X$ , where $A = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} u_2^T \\ u_1^T \end{bmatrix}$

(4) $Cov(Y) = \begin{bmatrix} \lambda_2 & 0 \\ 0 & \lambda_1 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$

(5) $\frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{3}{4} \longrightarrow$ 75% explained variance in 1-D projection

## Problem 6 – Cross-Validation

Suppose you have 100 training samples that you are using to train a classifier to distinguish between four classes. The training data has 50 samples of class 1, 25 samples of class 2, 20 samples of class 3 and 5 samples of class 4. To evaluate the stability and performance of your classifier on each class, you use 10-fold cross-validation. Is it a good strategy to randomly partition the data into 10 folds? Why or why not? If yes, fully justify why. If no, state why not, provide an alternate cross-validation scheme and justify the new scheme.

No, it is <u>not</u> a good idea to randomly partition the data as it contains an imbalanced number of points per class. We want to make sure we have sample representations in both training and validation.

Another strategy is, for example, perform cross-validation with a much smaller folder number maybe 3 or 4. For a 4-fold cross-validation, we can select randomly 10% of <u>each class</u> as the collective validation set (1 sample for class 4, 2 for class 3, 3 for class 2, and 5 for class 1) and the remainder of the samples (90% from each class) as the training set.

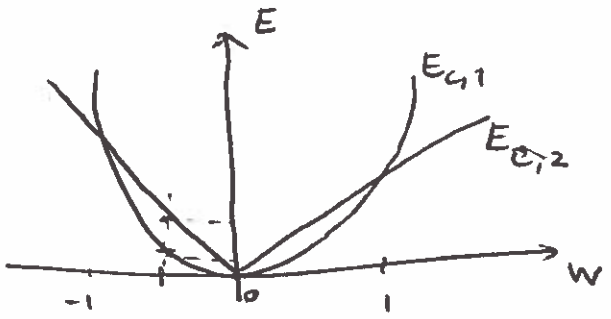
## Problem 7 – Regularization

Suppose you would like to perform feature selection by beginning with too many extracted features and, then, driving the weights for unnecessary features to zero using an appropriate regularization term. Considering the following two regularization terms,

$$E_{c,1} = \sum_{i=1}^{M} w_i^2$$

$$E_{c,2} = \sum_{i=1}^{M} |w_i|$$

which would be more effective for driving the weights to zero? Why? Clearly explain your reasoning.




The $L_1$ penalty ($E_{c,2}$) will be more efficient at driving the weights to zero, as it promotes sparsity in the model parameters.

In the penalty term $E_{c,1}$, the weight parameters are squared and, if the weigh are already small (less than 1) squaring them will make them even smaller and therefore $E_{c,1} < E_{c,2}$ for $|\vec{w}| < \vec{1}$.

Because $E_{c,1} < E_{c,2}$ for weights near zero but not zero, the penalty term will also be smaller, leading to a smaller cost value. This would lead $E_{c,1}$ to have ma weights near zero but not being driven to zero as quickl as if we used $E_{c,2}$.

## Problem 8 – ROC Curves and Confusion Matrices

Suppose you have the following training data set (associated with data collected during 1 minute of a detection system):

$$X = \{(1,1,2);\ (10,3,0);\ (-5,-4,1);\ (2,-3,1);\ (10,10,20);\ (0,0,0)\}$$

$$y = \{-1,1,1,-1,-1,1\}$$

where $y_i = 1$ indicates a true target and $y_i = -1$ indicates a non-target data point. Suppose you trained a classifier to produce a confidence of target given a sample. For the above data points, your MLP produced the following confidence values:

$$c = \{0.7,0.6,0.2,0.3,0,0.9\}$$

1. Draw the associated ROC curve.
2. To make a final decision, suppose you use a threshold at a confidence value of 0.5 where everything $\leq$ 0.5 is marked as non-target and $>$ 0.5 is marked as target. What would be your resulting confusion matrix?

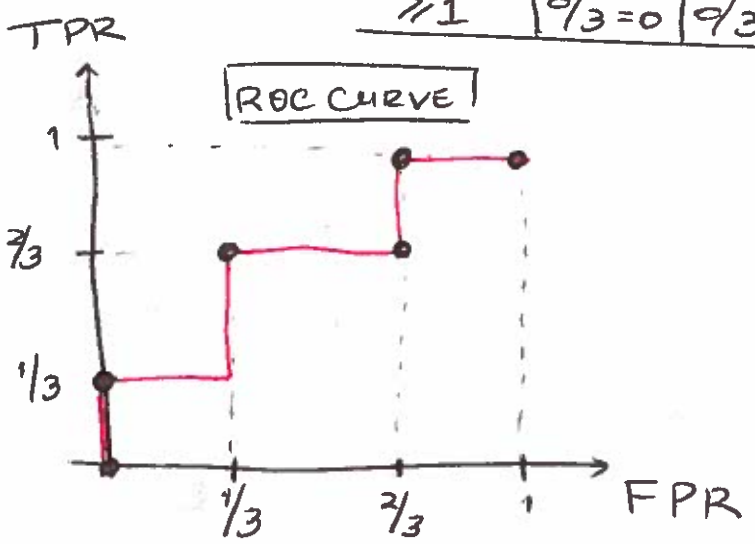
(i)

| Confidence | labels |
|---|---|
| 0.9 | 1 |
| 0.7 | −1 |
| 0.6 | 1 |
| 0.3 | −1 |
| 0.2 | 1 |
| 0 | −1 |

| Threshold | FPR | TPR |
|---|---|---|
| $\geq 0$ | $3/3 = 1$ | $3/3 = 1$ |
| $\geq 0.1$ | $2/3 \simeq 0.67$ | $3/3 = 1$ |
| $\geq 0.2$ | $2/3 \simeq 0.67$ | $3/3 = 1$ |
| $\geq 0.3$ | $2/3 \simeq 0.67$ | $2/3 \simeq 0.67$ |
| $\geq 0.6$ | $1/3 \simeq 0.33$ | $2/3 \simeq 0.67$ |
| $\geq 0.8$ | $0/3 = 0$ | $2/3 \simeq 0.33$ |
| $\geq 0.9$ | $0/3 = 0$ | $1/3 \simeq 0.33$ |
| $\geq 1$ | $0/3 = 0$ | $0/3 = 0$ |

$$FPR = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{TP + FN}$$



(2)

| | | Predicted values: 1 | Predicted values: −1 |
|---|---|---|---|
| True Labels | 1 | TP = 2 | FN = 1 |
| | −1 | FP = 1 | TN = 2 |

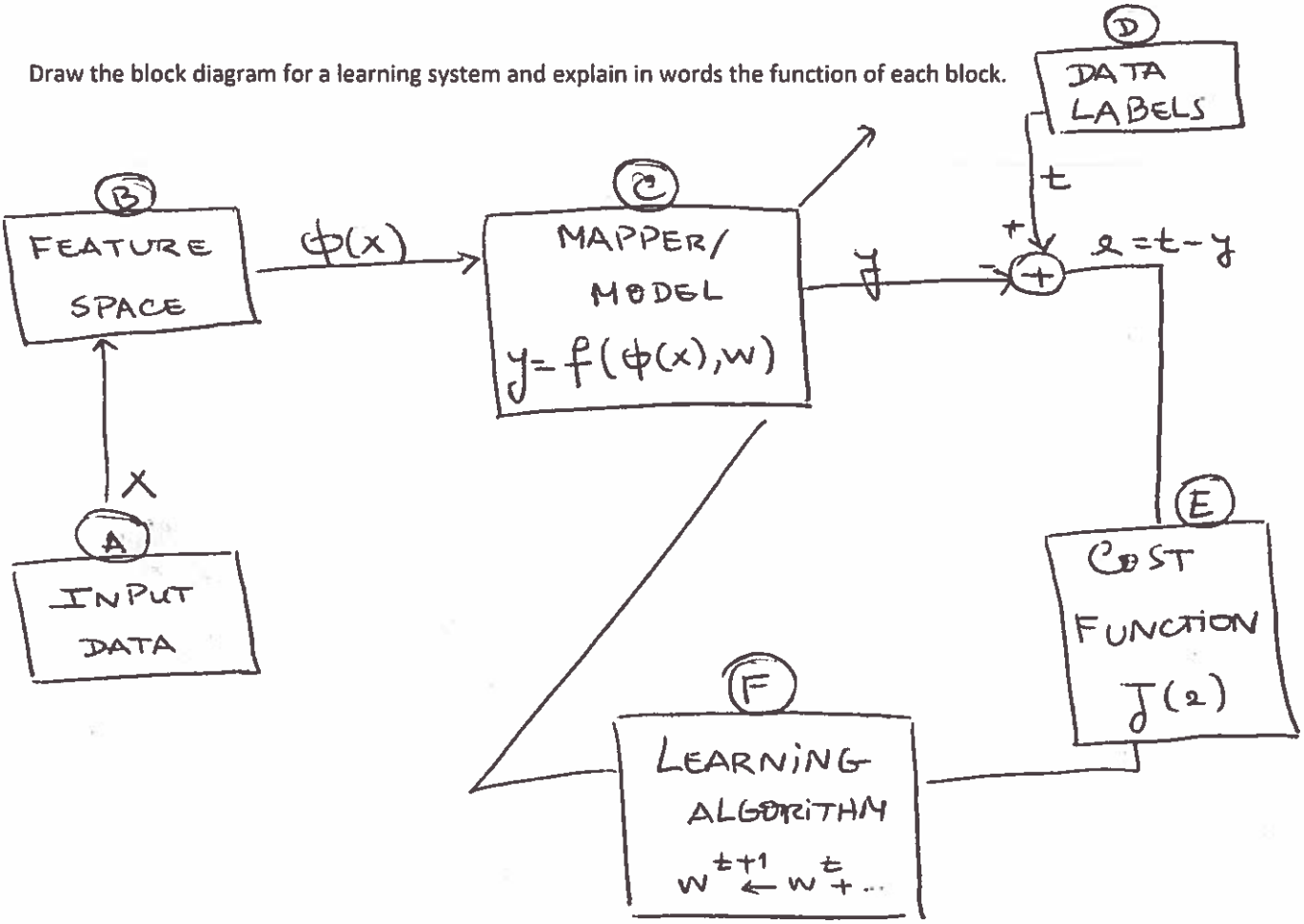
| True labels | sign f. | predicted |
|---|---|---|
| 1 | 0.9 | 1 |
| −1 | 0.7 | 1 |
| 1 | 0.6 | 1 |
| −1 | 0.3 | −1 |
| 1 | 0.2 | −1 |
| −1 | 0 | −1 |

$$FPR = \frac{FP}{FP + TN} = \frac{1}{3}$$

$$TPR = \frac{TP}{TP + FN} = \frac{2}{3}$$

## Problem 9 – Learning System

Draw the block diagram for a learning system and explain in words the function of each block.




A: INPUT DATA. DATA COLLECTION X.

B: FEATURE SPACE. FEATURE EXTRACTION / SELECTION, $\phi(x)$.

C: MAPPER/MODEL. Projects feature data with a set of trainable parameters. EXAMPLE: LINEAR REGRESSION, ANNs

D: DATA LABELS. Training labels, t.

E: COST FUNCTION: EVALUATES the ERRORS to allow selection of optimal weights. EXAMPLE: MEAN SQUARED ERROR.

F: LEARNING ALGORITHM: ACTUALLY changes the parameter to meet the design goal of minimizing ERRORS. EXAMPLE: gradient descent.

## Problem 10 – PCA vs Regression

What is the difference between PCA and regression? Be specific by explaining what each method accomplishes in terms of data representation. Illustrate with an example of a problem where you will use PCA and also another where you will use regression.

PCA is a procedure that finds an orthogonal coordinate system that is defined by the data. Does not require supervision. As such, it can be used to decorrelate the data and allows for subspace projections. One application is denoising and the other is ~~data~~ feature extraction.

Regression is a method to determine a manifold that passes through the data. (This manifold can be a curve ~~or~~, hyperplane or curved surface) It requires supervision. One application of regression is data prediction.

## Problem 11 – Dimensionality Reduction

Suppose you would like to use PCA to reduce dimensionality of your data prior to applying a $k$-NN classifier. Is PCA always an effective dimensionality reduction technique to be used in conjunction with classification? Why or why not?

Feature extraction is generally a good strategy to employ before classification. The reason for that is that we may not have enough data to describe the feature space.

PCA is as example of feature extraction. PCA works very well. However PCA is unsupervised and therefore the PCA projections may not preserve class separability and consequently pose a challenge during classification.

## Problem 12 – Decision Tree vs Random Forest

Describe (using pseudo-code and precise descriptions) the difference between a decision tree and a random forest. Focus your discussion on how the decision trees and random forests are constructed/trained.

Decision trees are non-parametric discriminative classifiers. Splitting nodes generally use two types of metrics: entropy or gini index. Decision trees make node splitting decisions that maximize the local criteria and therefore the final trained tree may not be the "optimal" tree. Decision trees are also prone to overfitting, and so random forests consider bootstrap samples from the original data and fit a decision tree to each. The final class decision is decided based on majority vote.

## Problem 13 – PCA vs LDA

Describe the differences between PCA and LDA. Illustrate each algorithm with an example.

Both PCA and LDA are feature extraction techniques. PCA performs dimensionality reduction in an unsupervised fashion — by preserving maximum variance of projections. LDA performs dimensionality extraction in a supervised fashion — directions of projection maximize class separability. LDA can also be used for classification.