CMPE 462 HOMEWORK 1

Due Date: 21.03.2019

In [ ]:

```
# Allowed libraries
import numpy as np
import matplotlib.pyplot as plt
# For loading data, you can import other libraries, but further usage is not allowed.
```

1.) Review bias/variance dilemma.

Replicate the bias-variance-plot (bias, variance and error vs. model complexity) given in Figure 4.6 in the textbook or in lecture notes [W04.pdf, pg.52]. For this:

- Generate 20 values $\{x^t\}$ from the range [0, 5] from uniform distribution.
- Generate 100 different samples X_s. Each sample contains 20 instances $\{x^t, y_s^t\}$ where $y_s^t = f(x^t) + \epsilon$, $f(x) = 2 \times sin(1.5 \times x)$, and $\epsilon \sim N(0, 1)$.
- For each sample, fit polynomial models of order 1,3, and 5.
- Plot bias, variance and error of these models.

In [ ]:

```
# Solution Here
```

2.) Review cross-validation.

Replicate the cross-validation error plot (training and validation error vs. model complexity plot) given in Figure 4.7 in the textbook or in lecture notes [W04.pdf, pg52]. For this:

- Generate 10 samples containing 100 instances as in part 1.
- Split each sample to training and validation sets, fit polynomial models of order 1,3, and 5.
- Plot mean training and validation error (mean square error) of each model.

In [ ]:

```
# Solution Here
```

3.) Use of real dataset.

Download Iris Data Set from UCI Machine Learning Repository. Randomly split it into training and test sets.

- For each feature, plot histogram of classes. For better visualiziation, plot classes in same plot but with different colors. (For this you can use plt.subplots)
- Consider only two classes: Iris Setosa and Iris Versicolour.
- Assuming Gaussian distribution, apply parametric classification, and find which particular feature (sepal length, sepal width, petal length, petal width) is most successful in classifying instances.
  - For this, treat each feature separately. find maximum likelihood estimate of the parameters (μ, σ^2) of different classes using training set. (You can use plt.errorbar for visualization, use one column for each feature and visualize both classes in same plot using different colors).
  - Using these parameters, predict the classes of the instances in the test set, and calculate the error.

In [ ]:

```
# Solution Here
```