CSE-5368 Neural Networks
Exercise Problems 02

Complete the following functions using only numpy:

```
def calculate_2d_mean_median_variance_columnwise(x):
    # given a two dimensional list of numbers.
    # Calculate mean, median and variance of each column
    # x: two dimensional list of numbers
    # return: two dimensional list of 3 rows by number of columns in x
    # The first row of the return list contains mean values,
    # 2nd row contains medians and third contains variances
    # Notes:
    # IMPORTANT:
    # Use only python. No other packages such as numpy, math, …
    # Assume that all the rows of x have the same number of elements

def calculate_2d_ mean_median_variance_rowwise(x):
    # given a two dimensional list of numbers.
    # Calculate mean, median and variance of each row
    # x: two dimensional list of numbers
    # return: two dimensional list.
    # the number of rows in the return list should be the same as
    # number of rows in x and return list should have 3 columns
    # Notes:
    # Use only python. No other packages such as numpy, math, …
    # Assume that the number of elements in each row are NOT THE SAME

def column_wise_normalization(x):

    # Write a function to normalize a numpy array, x,  column-wise,
    # by performing # the following calculation for each element:
    # (element value - mean of its column) / standard deviation of its column.

def mean_absolute_error(actual,desired):

    # This function calculates the mean absolute error
    # between two numpy arrays.

def confusion_matrix(actual, desired):

    # This function computes the confusion matrix for a multi-class
    # classification problem using actual and desired values.
    # actual is  a numpy array [input_dimensions,nof_samples]
    # desired is the same shape as actual


def covariance_matrix(X):

    # This function calculates the covariance matrix
    # for a given numpy array.

    # Covariance Matrix = ((X-X_mean)^T* (X-X_mean))
```