# Problem set guidelines

1. These problems require thought but do not require long answers. Please be as concise as possible.
2. Should you have any questions regarding this homework, please post them on Piazza or Slack. Chances are your classmates have them too. By posting and answering questions on Piazza or Slack, you help your peers to get better understanding of the class materials.
3. You can discuss the problems with your classmates but do not share your code or the answers and do not use someone else’s solutions.
4. Please submit the coding assignments in the form of Jupyter notebook along with the results of the execution and all the comments you like to make inline.
5. You can submit writing assignments in any form convenient for you. It could be LaTeX, MS Word, PDF or image. Please make sure it is of good enough quality.

## Technical details

1. Coding assignment use Python 3.6. You can use either [official distributive](https://www.python.org) or [Anaconda](https://anaconda.org/anaconda/python), which contains the majority of the required packages pre-installed for you.
2. Install packages from requirements.txt. You may need administrator right:

**pip3 install -r requirements.txt**

1. Please use [Jupyter](http://jupyter.org) for working with .ipynb files.In command line, type jupyter notebook.

# Prediction

**[40 points]**

In this problem you are going to build a model which predicts the unit price for distinct product. Data set for this task contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011. The company mainly sells unique all-occasion gift-ware.

**Attribute Information:**

*InvoiceNo:* Invoice number. Nominal. A 6-digit integral number uniquely assigned to each transaction. If this code starts with the letter 'c', it indicates a cancellation.

*StockCode:* Product (item) code. Nominal. A 5-digit integral number uniquely assigned to each distinct product.

*Description:* Product (item) name. Nominal.

*Quantity:* The quantities of each product (item) per transaction. Numeric.

*InvoiceDate:* Invoice date and time. Numeric. The day and time when a transaction was generated.

*UnitPrice:* Unit price. Numeric. Product price per unit in sterling.

*CustomerID:* Customer number. Nominal. A 5-digit integral number uniquely assigned to each customer.

*Country:* Country name. Nominal. The name of the country where a customer resides.

Take a look at **online\_retail.xlsx**. This file contains information for 2009-2010 and 2010-2011 years separately. Please load the data in your **Jupyter notebook** and investigate the features.

## 1a. Data preprocessing

**[10 points]**

Recall that ML works with numbers only. This dataset contains few non-numerical features. Present them in the numerical form before modeling (if needed).

## 1b. Preparing the subsets

**[10 points]**

In order to interpret the model performance, you need to split dataset into three subsets: train, cross-validation (CV) and test. Leaving 20% of data for cross validation and 20% for test will be ok for this task. Also, separate *UnitPrice* from the sets and store it in separate variables *y*. Recall that we want to train the model to predict price. It should not be a feature.

## 1c. Training regression model

**[20 points]**

Train regression model on the dataset. You do not have to write regression model on your own, use [scikit-learn implementation](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html). Please report how good your model is, given the CV and test set performance (use coefficient of determination (**R-squared**)).

# Classification

**[60 points]**

Now you got your hands dirty with training ML models for prediction. In this assignment, you will train classification models with much more real-world setup.

This problem deals with predicting breast cancer. Features in Data set are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

**Attribute Information:**

1) ID number

2) Diagnosis (M = malignant, B = benign)

3-32)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)

b) texture (standard deviation of gray-scale values)

c) perimeter

d) area

e) smoothness (local variation in radius lengths)

f) compactness (perimeter^2 / area - 1.0)

g) concavity (severity of concave portions of the contour)

h) concave points (number of concave portions of the contour)

i) symmetry

j) fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

All feature values are recoded with four significant digits.

Class distribution: 357 benign, 212 malignant

Take a look at **data.csv**. This is a raw data file. You can open in in any text editor or table editor such as Excel. CSV is one of the most popular data format files, you will often use in in your study and at work. It is worth to investigate how it looks in details.

Please load the data in your **Jupyter notebook** and investigate the features.

## 2a Exploring the data

**[5 points]**

Load the data from `data.csv`. If there are any missing values or non-numerical features, fix them.

## 2b Preparing the subsets

**[5 points]**

Split the dataset into three subsets: train, CV and test using 60-20-20% rule. Keep diagnosis separately as a label we want to predict.

## 2c Training logistic regression

**[15 points]**

Train logistic regression to classify the diagnosis. Use scikit-learn implementation of this model. Tweak the hyperparameters of the model to get the maximum performance on CV set.

## 2d Training SVM

**[20 points]**

Train support vector machines to classify the diagnosis. Use scikit-learn implementation of this model. Tweak the hyperparameters of the model to get the maximum performance on CV set.

Keep the model and the results separate from the logistic regression. You will need them for comparing the performance of the models.

## 2e Training Decision trees

**[10 points]**

Train Decision trees to classify the diagnosis. Use scikit-learn implementation of this model.

Keep the model and the results separate from two others. You will need them for comparing the performance of the models.

## 2f Check test set performance

**[5 points]**

Now, check the accuracy (use confusion matrix and classification report in scikit-learn) of all three models on the test set and compare it with the CV test accuracy. Explain the results.