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. Chances are your classmates have them too. By posting and answering questions on Piazza, 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 or Anaconda, 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
3. Please use Jupyter for working with .ipynb files. In command line, type jupyter notebook.

1 Prediction

[30 points]

In this problem you are going to build a model which predicts the exam score of a student given her demographics and historical records. The data was collected at two Portuguese schools to get a better understanding of the factors that influence students' progress.

The data and Jupyter notebook file with some pre-defined code for you can be found in the folder **Students**.

Take a look at **students_data.csv**. This is a raw data file, partially preprocessed for you. 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 open **Problem Set 1 - Students.ipynb** with Jupyter notebook, load the data and investigate the features. This part of code has already been written for you.

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.

1b. Preparing the subsets

[10 points]

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

1c. Training linear regression

[10 points]

Train linear regression model on the dataset. You do not have to write linear regression on your own, use scikit-learn implementation. Please report how good your model is, given the CV and test set performance.

2 Classification

[40 points]

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

This problem deals with predicting red wine quality. Picking up good wines takes professional sommelier a lot of knowledge and experience, they say. Let us see if ML can do this job.

This problem are the results of a chemical analysis of vinho verde wine samples, from the north of Portugal. Your goal is to predict wine quality based on physicochemical tests.

The data and Jupyter notebook file with some pre-defined code for you can be found in the folder **Wine**.

Please open **Problem Set 1 - Wine.ipynb** with Jupyter notebook.

2a Exploring the data

[5 points]

Load the data from `wine_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 wine quality separately as a label we want to predict.

2c Training logistic regression

[8 points]

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

2d Training SVM

[8 points]

Train support vector machines to classify the wine. 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 XGBoost

[9 points]

Train XGBoost to classify the wine. Use xgboost 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 two others. You will need them for comparing the performance of the models.

2f Check test set performance

[5 points]

Now, check the accuracy of all three models on the test set and compare it with the CV test accuracy. Explain the results.

3 Features and generalization

[30 points]

Imagine you are a spammer trying to break through the antispam filter and advertise your pills to every person in organization. You know that the organization uses antispam filer based on logistic regression with bag of words features.

Bag of words represents every email as a single vector. The vector indicates if every word of English language is present in the email. Typically, bag of words vector for English is about 50000 elements long. For example, the sentence “a cat and a dog” will be represented as the following:

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \vdots \\ \text{and} \\ \vdots \\ \text{cat} \\ \vdots \\ \text{dog} \\ \vdots \\ \text{zygmurgy} \end{matrix}$$

3a Hacking the spam filter

[15 points]

How would you put together an email which would still advertise your pills but will make it through this spam filer?

3b Improving the spam filter

[15 points]

Now, how would you improve the logistic regression model so that the spammer cannot hack your antispam system so easily?