UCL DEPARTMANT OF MEDICAL PHYSICS AND BIOMEDICAL ENGINEERING

Undergraduate / Postgraduate
Assessed Coursework Tracking Sheet

Module Code:	MPHY0041
Module Title :	Machine Learning in Medical Imaging
Coursework Title :	Assessed Coursework
Lecturer:	Dr. Andre Altmann
Date Handed out:	Friday, February 4th 2022
Student ID (Not Name)	

Submission Instruction: Before the submission deadline, you should digitally submit your source code and generated figures (e.g., a single jupyter notebook file is ideal) and a PDF file containing your written answers (alternatively, write answers in the jupyter notebook). All files need to be combined in one single zip file and submitted on the module page at UCL Moodle.

Coursework Deadline:	Friday, **March 4th 2022 at 16:00** at UCL Moodle submission section
Date Received	
Date Returned to Student:	

The Department of Medical Physics and Biomedical Engineering follows the UCL Academic Manual with regards to plagiarism and coursework late submission.
UCL Policy on Plagiarism
UCL Policy on Late Submission of Coursework

If you are unable to submit on-time due to extenuating circumstances (EC), please refer to the UCL Policy on Extenuating Circumstances and contact our EC Secretary at medphys.teaching@ucl.ac.uk as soon as possible.
UCL Policy on Extenuating Circumstances

Please indicate what areas of your coursework you particularly would like feedback on:

Mark (%):

Please note that the mark is provisional and could be changed when the exam boards meet to moderate marks.

Please note: For the derivation of decision boundaries, please show your reasoning/workings in addition to the final formula for the boundary. Feel free to use a computer to derive matrix multiplications, matrix inversions, matrix determinants, etc. Preferably, please submit a single jupyter notebook file for Exercises 1, 2 and 4. The file should contain code, plots and comments that help the understanding of your answers (you can give your answers as a Markdown in the jupyter notebook). If you are not using a jupyter notebook, then please store the figures make sure that they can be easily attributed to the corresponding part in the code.

1. Load the dataset `'PPMI_DATSCAN.csv'` it contains data from Dopamine Transporter Scan (DaT scan) of two brain regions in the left and right hemisphere of the brain: caudate and putamen. DaT scan is a single-photon emission computed tomography (SPECT) method to measure the loss of dopaminergic neurons in diseases such as Parkinson's disease. It also contains the values of a test for motor abilities (MDS UPDRS Part II). The dataset comprises Healthy Controls (HC), Parkinson's disease (PD) and Scans Without Evidence of Dopaminergic Deficit (SWEDD). The column 'COHORT_DEFINITION' denotes the diagnosis.
    a) Remove SWEDD subjects from the dataset. Compute means for DaT scan in the left putamen (DATSCAN_PUTAMEN_L) for the 'HC' (μ_{HC}) and the 'Parkinsons' (μ_{PD}) groups. In addition, compute the standard deviation (σ) for DaT scan in the left putamen in that dataset. Assume that the data follow a Gaussian distribution:
    $$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2},$$
    with the means and standard deviations as computed above. Compute the decision boundary between the two disease groups (with the prior probabilities $\pi_{\mathrm{HC}} = \pi_{\mathrm{PD}} = 0.5$).
    (In case you cannot compute means and standard deviation from the data, to complete the remainder of the exercise, you can use $\mu_{\mathrm{HC}} = 2.1$, $\mu_{\mathrm{PD}} = 0.8$ and $\sigma = 0.7$.) [6]

    b) Using `sklearn` functions, train a `LinearRegression` to separate HC from PD subjects using both DaT scan values for the left putamen and MDS UPDRS Part II score (NP2PTOT) inputs. Generate a scatter plot for DaT scan in the left putamen and NP2PTOT using different colours for two diagnostic groups. Compute the decision boundary based on the linear regression and add it to the plot. [4]

    c) The previous analyses ignored the subjects with SWEDD. Going back to the full dataset, compute means for all three groups for DaT scan in the left putamen and the NP2PTOT score as well as the variance-covariance matrix Σ (Hint: Use vector representations, the means have dimension 2x1 and Σ has dimension 2x2). Use these to compute linear decision boundaries between all pairs of classes (with the prior probabilities $\pi_{\mathrm{CN}} = \pi_{\mathrm{MCI}} = \pi_{\mathrm{Dementia}} = 0.33$). Generate a new scatterplot and add the three decision boundaries.

(In case you cannot compute means and covariance matrix from the data, to complete the remainder of the exercise, use $\mu_{\mathrm{HC}} = (2.1\ \ 0.4)^T, \mu_{\mathrm{SWEDD}} = (2.0\ \ 5.5)^T, \mu_{\mathrm{PD}} = (0.8\ \ 6.5)^T$ and $\Sigma = \begin{pmatrix} 0.5 & -1.4 \\ -1.4 & 24.5 \end{pmatrix}$.) [6]

2. Here we fix different implementations for Logistic Regression. One using the iterative weighted least square solution (lecture slide set 15) and the other using gradient descent (lecture slide set 12).
   a) The file `LogReg_IWLS_gaps.py` contains a few gaps that need to be filled for the function to work. Complete the function by computing the W matrix (lines 23 and 40) the logLikelihood (lines 27 and 43), the z vector (line 32) and the new set of betas (line 35). Use your function to train a model that separates Healthy controls from PD subjects in the `LogReg_data.csv` dataset (PD column indicates PD status, remaining columns are the features) and provide the estimated beta coefficients.

      (Hint2: The operator `@` can be used for matrix multiplications; the function `np.linalg.inv()` computes the inverse of the matrix X^{-1}). [5]

   b) The file `LogReg_GRAD_gaps.py` aims to implement Logistic Regression using gradient descent (lecture slide set 12). However, there are still a few gaps in the code. Complete the computation of the cost (`J(β)` in the slides) in lines (25 and 35) as well as the update of the beta coefficients (line 29). (Hint: gradient descent aims to *minimise* the cost; however, Logistic Regression is fitted by *maximising* the log likelihood). Use your function to train a model that separates Healthy controls from PD subjects in the `LogReg_data.csv` dataset and provide the estimated beta coefficients. [5]

   c) Extend your `LogReg_GRAD_gaps.py` from b) to implement Logistic Regression with a L2-regularisation on the beta coefficients (i.e., like Ridge Regression, but for Logistic Regression). (Note: the function will need a new parameter: λ). Run your algorithm on the data from a) with regularization parameters $\lambda = 0, 1, 10$ and provide the resulting beta coefficients. [5]

3. A collaborator hands over to you a dataset with data on *N*=300 subjects and *p*=10000 features (various continuous health readouts and biomarkers such as blood pressure, sleep duration, weekly exercise hours, …). They would like you to build a model that predicts a continuous measure corresponding to 'healthy aging' which was assessed using a questionnaire. Briefly describe your analysis plan (i.e., Any data pre-processing? Feature Selection? Which learner(s) would you consider and why? What are the necessary steps to quantify how well your model works?). [8]

4. In this exercise you are using real data from the Parkinson's Progression Marker Initiative (PPMI) Database (https://www.ppmi-info.org/). PPMI is a multicenter initiative that acquires longitudinal data from people with Parkinson's disease, healthy controls and people with SWEDD. PPMI makes the data freely available to any interested researcher. PPMI's aim is to identify sensitive biomarkers for the disease. Here, you are dealing with measurements of subcortical volumes (`'LLatVent'` to `'Raccumb'`), cortical thickness (ending in `'_thickavg'`), cortical surface area (ending in `'_surfavg'`) and cortical volumes (ending in `'_volume'`) for different brain regions derived from T1-weighted magnetic resonance imaging (MRI) using the FreeSurfer software. The provided file also contains clinical diagnosis, DaT Scan measures, cognitive measures etc.

   TASK: Researchers would like to assess whether it is possible to predict PD status from the T1 weighted MRI scans. Thus, this machine learning task aims to use different classification methods to predict the disease label.
   DATA: Use `PPMI_CW1_TRAIN.csv` for training and `PPMI_CW1_TEST.csv` for testing. In addition to the brain MRI features, the files also contain the diagnosis (`DX`), the subjects' sex (`Sex`), age of the participant (`Age`) , a cognitive measure (`MoCA`), intracranial volume (`ICV`) and DaT scan for four brain regions.

   Use functions from `sklearn` to train classifiers and make predictions. Use as inputs the MRI brain measures and as target the diagnosis `DX`. In this setting we are interested in the Area Under the ROC Curve as the performance measure.

   a) Train a `LogisticRegression` with elastic net penalty with `l1_ratio=0.5`, optimize C using a 10-fold cross-validation. Provide plots on how the performance develops on training, testing and cross-validation in response to C. What is the performance of your final fitted model on the test data? [6]

   b) Train a support vector classifier (`SVC`) considering the linear kernel, polynomial kernel (degree=3) and the `rbf` kernel. Tune the cost-parameter and kernel choice using 10-fold CV. Provide a plot on how the performance of the SVR evolves depending on C and kernel choice on training, test and cross-validation errors. What is the performance of your fitted model on the test data?
   (Hint: when SVC seems to take an endless time to train, then change your choice of C parameters; large C parameters = little regularization = long training time. Use the function `np.logspace()` to create a series of C parameters, e.g., `np.logspace(-4,2,base=10,num=50)`). [6]

   c) Train a `RandomForest` classifier (`min_samples_split=5` and 500 trees). Provide performance for training, testing and the 'out-of-bag' estimate. [4]

   d) To provide estimates of your model's performance on unseen data, set up a nested cross-validation with 10 outer folds and 5 inner folds (you can use `sklearn` commands for that). Use this to compare the performance for `LogisticRegression`, `SVC` and `RandomForestClassifier` (using the suggested settings from a)-c); e.g., elastic net penalty for Logistic Regression etc.). What is the mean and standard deviation of the models' performance? [5]

e) From the T1-weighted MRI the model uses four groups of features: subcortical volumes, regional cortical thickness, regional cortical surface area and regional cortical volumes. Using the LogisticRegression with the elastic-net penalty and cross-validation, which set of features performs best and does using all features work better than any single set? Can the advantage be carried over to the test set performance? [5]