Assignment 3: Supervised Machine Learning

The goal of the assignment is to help you get familiar with feature engineering and the use of machine learning toolkit (scikit-learn) for constructing supervised machine learning models to solve real problems. This assignment has three tasks: regression, multi-category classification, and multi-label classification (i.e., tagging). In each task, you will explore and then select two different models, and compare their performance on the given dataset. You will also write a report to record what you have done in the experiments.

Dataset: Student Portuguese Class Performance Data Set, altered version of URL.

Description: This data set contains student achievements in secondary education of two Portuguese schools.

Training, validation and testing: 585 students data in assign3_students_train.txt will be used as training and validation data; 64 students data in assign3_students_test.txt will be used as testing data.

Attributes:

1. school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2. sex - student's sex (binary: 'F' - female or 'M' - male)
3. age - student's age (numeric: from 15 to 22)
4. address - student's home address type (binary: 'U' - urban or 'R' - rural)
5. famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6. Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
7. Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
8. Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
9. Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10. Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11. reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12. guardian - student's guardian (nominal: 'mother', 'father' or 'other')
13. traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14. studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15. failures - number of past class failures (numeric: n if 1<=n<3, else 4)

16. edusupport - student receive extra educational support (nominal: 'school' (extra educational support from school), 'family' (from family), 'paid' (extra paid Portuguese classes) or 'no' (no extra educational support))
17. nursery - attended nursery school (binary: yes or no)
18. higher - wants to take higher education (binary: yes or no)
19. internet - Internet access at home (binary: yes or no)
20. romantic - with a romantic relationship (binary: yes or no)
21. famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
22. freetime - free time after school (numeric: from 1 - very low to 5 - very high)
23. goout - going out with friends (numeric: from 1 - very low to 5 - very high)
24. Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
25. Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
26. health - current health status (numeric: from 1 - very bad to 5 - very good)
27. absences - number of school absences (numeric: from 0 to 93)
28. G3 - final grade (numeric: from 0 to 20, output target)

Task 1: Predict student final grading with regression models.
This is a regression task. In this task, you will predict the student final grade (attribute 28) with a regression model. You can use all or a subset of attributes 1-27 as features, and even build new features based on these attributes. Models that can be selected from, but not limited to, include Linear Regression, Support Vector Regression, Decision Tree Regression, Nearest Neighbor Regression. You will use **Mean squared error (MSE)** as the evaluation metrics.

Task 2: Predict the student mother's job with classification model.
This is a multi-category classification task. In this task, you will predict/classify a student mother's job (attribute 9) to one of the possible values: teacher, health, service, at_home, other. You can use all or a subset of the rest attributes (1-8, 10-28) as features, and even create new features. Models that can be selected from, but not limited to are: Logistic Regression, Support Vector Machine, Decision Tree, Nearest Neighbor, Naive Bayes. You will use {Accuracy, Precision, Recall, F1} as the evaluation metrics. You will need to report **{Precision, Recall, F1} for each category**, and also report **macro-{Precision, Recall, F1} and accuracy for the whole dataset**.

Task 3: Predict what kind of extra educational support the student may receive.
This is a multi-label classification task, which is also called tagging. Students may receive extra educational support from school, family, and paid classes. In this task, you need to predict attribute 16. Please note that one student may have one or several values in this attribute (that is why we call it tagging, just like a webpage can have one or many tags). For example, one student may have "no" extra educational support, while another student may have received

school, family and extra paid support. Therefore,multi-label classification is usually used as a tagging model.

**** Please differentiate multi-category classification from multi-label classification. If you wrongly design this task as a multi-category classification task, there will be heavy points deduction.**

You can use all or a subset of the rest attributes (1-15, 17-28) as features, and even create new features. You may want to use the One-vs-the-Rest strategy to get multiple labels. Classification models that can be selected from, but not limited to, are Logistic Regression, Support Vector Machine, Decision Tree, Nearest Neighbor, Naive Bayes. You will use **Accuracy and Hamming loss** as the evaluation metrics.

NT: Hamming loss is defined in https://scikit-learn.org/stable/modules/generated/sklearn.metrics.hamming_loss.html#sklearn.metrics.hamming_loss, and google more information online, such as https://www.youtube.com/watch?v=Dn-w2favLHw.

For each task, you need to:

- Choose two classification models,
- Tune the model parameters and the construction of your features based on the training data with 10 fold cross-validation.
- Learn a model with your best parameters and features and test it with testing data.
- Write the report in **report.txt**:
    - Copy your output of the main function.
    - In terms of features, you need to answer:
        - What features do you choose to use, and why do you choose them?
        - How do you use these features (feature engineering)? For example, use its original value, normalized value, log value, one hot vector, etc.
    - For each model, you need to write:
        - The model name.
        - What parameter you have tried, and corresponding performance on training data?
        - Final performance of learned model on the testing data.
        - How long to train the model on training data.

Reference materials:
Classification models: https://www.youtube.com/watch?v=ppXFoItcX7A
Regression models: https://www.youtube.com/watch?v=ZkjP5RJLQF4
Cross validation: https://www.youtube.com/watch?v=fSytzGwwBVw
Evaluations metrics:
https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics
For more, just "**google it**". There are plenty of online materials. Learn by yourself.

Tools:
Scikit-learn(https://scikit-learn.org/stable/index.html#).
If you prefer to use other tools, please get permission from TA before doing assignments.

Coding:
Please feel free to add extra python files, functions, attributes to do training, validation, and testing. Your grading will be based on the output of running Assignment3Main.
Assignment3Main is the main class for running your assignment, which **cannot** be modified. It calls **model_1_run() and model_2_run()**, where you first implement train the model on training data with your best parameter obtained in validation, and then evaluate and output its performance on testing data. You need not submit the codes generated in the tuning procedure.

******Extra bonus****:**
We will run a small competition for task 3 submissions. Top 10 (the better performed one of the two models) submissions (out of 39 students) will receive **extra 10 points**. Evaluation metrics are Accuracy and Hamming loss. After collecting all submissions, we will publish the ranking and the reports of the top 10 runs.

Grading:
Your submission will be graded based on:

1. Please make sure that your program is runnable. If not, -50%.
2. Correctness of the implementation on three tasks (45%)
3. A clear report in report.txt (20%)
4. A reasonable tuning procedure of hyper parameters (10%)
5. A reasonable selection of feature set (10%)
6. Learned model's performance on testing data (15%)

Top 10 runs on Task 3 will receive extra 5 points.

Submission Requirements

A zipped file package with the naming convention as "pittids_a3". For example, suppose the Pitt id is jud1, then the submission package should be jud1_a3.zip.
The file package should contain:

1. All the scripts/programs you used for this assignment. (**src folder**)
2. A clear report with the output of Assignment3Main. (**report.txt**)

Do not upload the assign3_train.txt and assign3_test.txt.