ISB46703 Principles of Artificial Intelligence
Universiti Kuala Lumpur
September 2022
Dr. Faiz

Mini Project

Weight: 15%
Due: 31 December 2022

1 Introduction

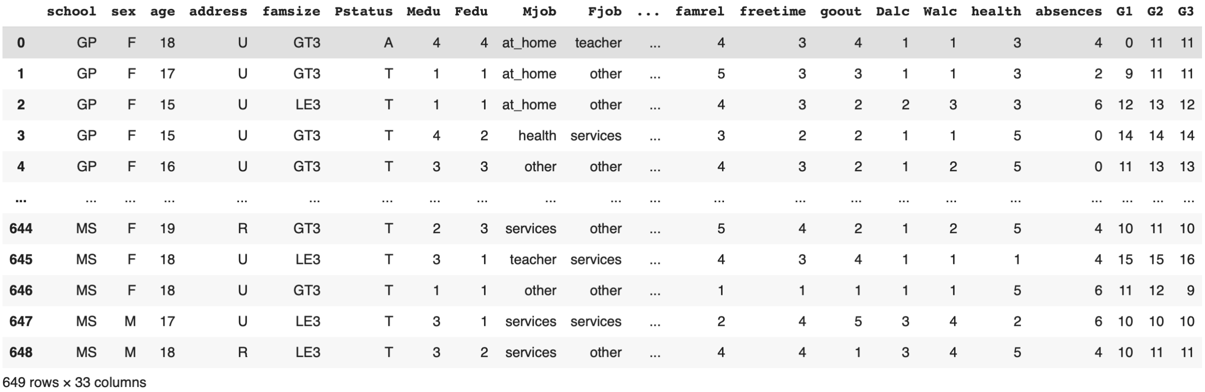
In this project, you are going to build a classifier to predict student performance using students' past performance data. You will use the student performance dataset, which is available on the UC Irvine machine learning repository at `https://archive.ics.uci.edu/ml/datasets/student+performance`

1.1 Goal

Your final goal is to predict whether the student has **passed** or **failed**.

2 Dataset

The dataset contains the data of about 649 students, with and 30 attributes for each student. The attributes formed are *mixed* categorically between word and phrase, and numeric attributes. **These mixed attributes cause a small problem that needs to be fixed** using **one-hot encoding** by utilising Pandas `get_dummies()` function. Figure 1 shows the first and last 10 attributes from the data.



	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	4	0	11	11
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	2	9	11	11
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	6	12	13	12
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	0	14	14	14
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	0	11	13	13
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
644	MS	F	19	R	GT3	T	2	3	services	other	...	5	4	2	1	2	5	4	10	11	10
645	MS	F	18	U	LE3	T	3	1	teacher	services	...	4	3	4	1	1	1	4	15	15	16
646	MS	F	18	U	GT3	T	1	1	other	other	...	1	1	1	1	1	5	6	11	12	9
647	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	6	10	10	10
648	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	4	10	11	11

649 rows × 33 columns

Figure 1: *Students Performance* dataset.

Some of the attributes are categorical, such as the name of the `school`, `sex`, `Mjob`; which is the mother's occupation and `Fjob`; which is the father's occupation. Others, such as `age` and `freetime`, are numeric.

3 Test Scores

The dataset has three test scores: `G1`, `G2`, and `G3` (out of possible 20). Rather than taking the sum of these scores, **you will need to simplify the problem** by just providing pass (sum of `G1`, `G2`, and `G3` >= 35) or fail (sum of `G1`, `G2`, and `G3` < 35). In other words an **additional column (attribute) called `pass` needs to be added** to the dataset; whose value is either 0 == *fail* or 1 == *pass*.

4 Modelling

4.1 Training and Testing

Split the training and testing data using 70 : 30 ratio using `train_test_split()` function. Use all attributes (columns) as input X whilst `pass` as output (label) y, resulting in training input and label pair (`X_train`, `y_train`) and testing input and label pair (`X_test`, `y_test`).

4.2 Feature Scaling

Scale all features prior to building the classifier; (`X_train`, `X_test`) with a zero mean and unit variance, altogether by computing its z–score:

$$z = \frac{x - \mu}{\delta}$$

Use `fit_transform()` function from `StandardScaler()` module to scale the features.

4.3 Classifier

Build two **Support Vector Machine** (SVM) classifiers using `SVC()` function from the `scikit_learn` package. Perform the *pass* and *fail* classification using:

1. Linear SVM (`SVC(kernel="linear")`)
2. Non–linear SVM (`SVC(kernel="rbf")`) with *radial basis* kernel function (RBF).

Report the classification performance using the following metrics:

1. Accuracy
2. Precision
3. Recall
4. F1-Score

Plot the classification confusion matrix using `ConfusionMatrixDisplay()`.

4.4 Cross–Validation

Rebuild the classifiers (linear and non-linear SVMs) using 5–fold cross validation. **Plot the accuracy for each fold and report the mean and standard deviation accuracy**.

4.4.1 Parameters tuning

For the non-linear SVM classifier, repeat the k-fold cross-validation ($k = 5$) to find the optimal C and gamma γ parameters combination (grid search) from the following range:

1. gamma γ: $10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$
2. C: $10^{-1}, 1, 10, 10^{2}, 10^{3}$

Grid search can be done using `GridSearchCV()` function from `sklearn` `ModelSelection` library. Perform classification using the optimal parameter and report the performance using the following metrics:

1. Accuracy
2. Precision
3. Recall
4. F1-Score

Plot the classification confusion matrix using `ConfusionMatrixDisplay()`.