Exercise 6: Python: Object oriented programming

In this exercise you will create another library and practice object oriented programming and the use of classes. For this exercise divide yourself in groups of 2-3 students (same or different from previous homeworks).

For this project each of the components of the group should do at least one commit. Pay attention to the commit messages and try to follow the best practices seen in class.

You need to do at least one Pull Request per group.

1. Create a new virtual environment for this project.
2. Create a repository in the account of one of the components of the group.
3. Clone the repository locally each of the components of the group.
4. Create a Python library following the structure seen in class. (Needless to say, use different names to the folders and files that represent are representative of the purpose of the code they contain).
5. Complete the exercises in hw6.py file and commit the solutions.
6. Finally, you are going to adapt the data analysis done during the hw5 on the data from "sample_diabetes_mellitus_data.csv". As in hw5, imagine you are in a project that tries to predict whether some patients have diabetes mellitus. In order to do so, you create a library that will be used for your team to perform the analysis and make predictions. Since you want to collaborate effectively with them, you will create a bunch of classes that will perform each of the steps of the process following the best practices learnt during the last classes. Try to create interfaces for each of the important parts of the project, such as load/read, feature extraction and modeling. Separate the classes into files/folders in a way that makes the most sense to the team. For the purpose of this exercise, you are going to follow the next steps:
    a. Create a class with a primary method that loads the data and returns two dataframes, one for train and another for test. Internally, the class can use the function defined in hw5.
    b. Create a preprocessor class that removes those rows that contain NaN values in the columns: age, gender, ethnicity.
    c. Create a preprocessor class that fills NaN with the mean value of the column in the columns: height, weight.
    d. Create at least two feature classes that transform some of the columns in the data set. These feature classes need to have the same structure defined by an abstract parent class (Remember: polymorphism).
    e. Create a model class with two primary methods: train and predict. When the model class is initialized, the constructor (**init**) should receive as inputs (at least):
        1. Feature columns that are going to be used
        2. Target column that is going to be used
        3. (bonus) Hyperparameters of the model to be used.
    f. The model class should have as private attributes each of the inputs of the constructor and an additional one, called "model" that will be a model from sklearn chosen by the team (such as LogisticRegression or RandomForestClassifier) as a public attribute of the class.
        1. The train method should receive the train data containing at least the feature and target columns defined and fit the self.model on the train data using the features and the target (to filter columns) passed when the class is initialize, and return nothing. 2.The predict method should receive a dataframe, use the features passed to filter the columns and return the predicted probabilities using the .predict_proba method of the sklearn class selected.
7. Create a notebook that imports the classes and executes the previous steps in order to load, preprocess, create the features, train and predict the probabilities on the test set. Add a new column in the test set called predictions and compute the roc_auc_score as you did in hw5. Notice that the purpose of the exercise is to practice programming skills, the actual results on the metric, the appropriateness of the split, etc. are not going to be evaluated.
8. Share the link to the repository by turning in the assignment and let my user (Icedgarr) have access to it.