CS5228 Lab Assignment 1
Classification

Semester 2 2018/19
Due date: February 22, 2019

In this assignment, you are a data scientist at the Pima Womans Hospital of Arizona. You have been asked by the hospital to develop a computer program that can accurately diagnose whether a patient has diabetes. You are provided with a dataset containing historical records of measurements and outcomes. You will be using some common data mining libraries to achieve your objective. Detailed instructions and questions are given in following sections.

An IPython Notebook file, AssignmentOne.ipynb, is provided. You are required to install Jupyter Notebook in order to open and edit the file. You are expected to set up the programming environment and install all the libraries on your own – how to install and use off-the-shelf data mining packages is also part of the learning. For this assignment, you will be using:

- Python version 3.6 or later
- Jupyter Notebook
- scikit-learn 0.20.2
- pandas
- NumPy
- Matplotlib
- seaborn

These libraries are imported at the beginning of the AssignmentOne.ipynb file. You are allowed to use any additional libraries. Be sure to import them at the beginning of the IPython notebook.

Submission

You are expected to write all your codes and answers within the indicated spaces in the IPython notebook (answers to the conceptual questions can be embedded in the Notebook as markdown cells). **You should not delete or modify any of the existing codes or texts. Prior to submission, run all the cells to generate and save the results and answers.** We will grade your assignment based on the generated results. Submit a single IPython notebook with the name "YourNameInIVLE_YourIDINIVLE.ipynb" to the submission folder in IVLE.

(If you have further questions, you can email *ruidanhe@comp.nus.edu.sg*.)

The Data

The Pima Indians of Arizona have the highest rate of obesity and diabetes ever recorded. This dataset was collected by the National Institute of Diabetes and Digestive and Kidney Diseases, with the objective to predict whether or not a patient has diabetes based on certain diagnostic measurements. All the patients recorded in this dataset are females of the Pima Indian heritage. The dataset consists of 8 medical predictor variables and one target variable, *Outcome*. Table 1 shows the description of each variable. Your task is to build classification models for predicting *Outcome* based on other variables.

Pregnancies	Number of times pregnant
Glucose	Plasma glucose concentration in an oral glucose tolerance test
BloodPressure	Diastolic blood pressure (mm Hg)
SkinThickness	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (muU/ml)
BMI	Body mass index (weight in kg / (height in m)^2)
DiabetesPedigreeFunction	Diabetes pedigree function
Age	Age (years)
Outcome	Class variable (0 or 1)

Table 1: Dataset description

What to Do

Tasks are divided into 4 parts: 1) *Data Analysis* ⟶ 2) *Feature Transformation* ⟶ 3) *Model Construction* ⟶ 4) *Best Model Construction*. Instructions and their corresponding marks allocated are given in each part.

1 Data Analysis *(15 points)*

Load the CSV file into a DataFram object named ***pima*** using Pandas to examine the dataset and perform data cleaning:

1) Print the dtype of each column to review the data types of the attributes in the dataset. *(2 points)*

2) Are there any missing values in the dataset? Print out the total count of missing values for the attributes in the dataset. How would you handle the missing values without deleting any record? Develop and execute an operation for that purpose accordingly. *(3 points)*

3) Examine *Outcome* by generating a bar plot showing the count of "0" and "1" in *Outcome*. Write down any of your insights from observing the bar plot. *(3 points)*

4) Generate a pairwise relationship scatterplot (8x8 subplots) using
sb.pairplot(pima, vars=feature_column_names, hue='Outcome'). *(2 points)*

5) Generate a 9x9 heatmap plot showing the co-variance between any two features (including target). *(2 points)*

*(Hint: use **sb.clustermap(pima.corr(),annot=True)** to plot co-variance heatmap.)*

6) What are the insights you gained from the two plots generated from 4) and 5)? *(3 points)*

(Hint: think about how to perform feature selection based on the observations, which may be helpful for constructing your best model in Section 4.)

2 Feature Transformation *(15 points)*

We observe that all the variables in the dataset, except the target variable *Outcome*, are continuous. In this section, you will perform feature transformation on the continuous variables to generate two sets of features that will be used in the subsequent classification task.

Feature set 1 – Continuous to Categorical.

1) Often when dealing with continuous variables like *BMI* or *BloodPressure*, we may wish to transform these continuous variables into categorical variables which may be better predictors of *Outcome*. This transformation can be achieved as follows: cut the continuous values of each feature into non-overlapping buckets. Perform this operation on all 8 continuous variables. You are to devise your own method to cut the continuous values into buckets. Name the resulting DataFrame as ***pima1***. pima1 should still contains 9 columns with the same set of column names as pima. Use ***pima1.head()*** to show the top rows. *(5 points)*

*(Hint: you can use **pandas.cut()** or **pandas.qcut()** to convert continuous feature to categorical feature. e.g. For feature Glucose, if you wish to adopt the glucose level guidelines given by American Diabetes Association, you can convert its values into 3 groups:*
- Normal >140mg/dl of glucose,
- Prediabetes 140-199mg/dl of glucose,
- Diabetes >200mg/dl of glucose
*using **pd.cut(pima['Glucose'],[50,139.99,199.99,250], labels=[0, 1, 2])**. Or you can use **pd.qcut(pima['Glucose'], 4, labels=[0,1,2,3])** to cut the values according to quantile-based discretization function.)*

2) Next, we convert the generated categorical features into binary features using the **one-hot encoding scheme**. Assume the continuous feature has *m* labels. The one-hot encoding scheme will results in a vector of size *m* with only one of the values as 1 (indicating it as active). Use ***pima1.head()*** to show the top rows of the encoded pima1. *(2 points)*

*(Hint: you can use **pd.get_dummies()** to convert each categorical feature in pima1 into multiple binary features.)*

3) Discuss whether the use of one-hot encoding scheme can be omitted and why? *(3 points)*

Feature set 2 – Features Normalization

4) For the second feature set, we normalize the values of 8 continuous variables. For each variable, we apply the transformation using the formula $z = \frac{x-u}{s}$, where u and s are the mean and standard deviation of the variable values. Name the resulting DataFrame as ***pima2***, and show the top rows using ***pima2.head()***. *(2 points)*

*(Hint: you can use **StandardScaler** from scikit-learn for standardization purpose)*

5) Briefly discuss whether Feature set 1 or Feature set 2 is more useful for training classification models and why? Also comment on whether feature normalization is necessary in this case. *(3 points)*

3 Model Construction *(55 points)*

Your task in this section is to use the derived feature sets from previous section to construct classification models for diabetes outcome prediction (0 or 1). We have partitioned the data into training and test sets for you in the IPython notebook.

1) Experiment with the 5 classification models below from scikit-learn with their default hyperparameter settings on pima1 and pima2 (you should thus perform 5x2 times model training and evaluation):

- Gaussian naive bayes
- K-NN
- SVM
- logistic regression
- decision tree

You should perform model training on *x_train1* and *x_train2*, and perform evaluation on *x_test1* and *x_test2* respectively. Use **accuracy** and **weighted F1** as evaluation metrics, and save the results of different models in lists *pima1_acc*, *pima2_acc*, *pima1_f1*, *pima2_f1*. Note that the order of results in each list should match the model order provided in *model_names*. Print the *accuracy_record* table and *F1_record* table using the provided code. *(20 points)*

2) According to the results above, which feature set is better, pima1 or pima2? Select one to be used in the following tasks. *(2 points)*

3) Select one classification model from 1). Discuss which are the hyperparameters that may affect the model performance the most. Perform grid search with 10 folds cross-validation for tuning those hyperparameters on the training set of either pima1 or pima2 (according to your choice in 2)), using accuracy as the scoring metric. Print the configuration of the best selected model and its prediction **accuracy** and **weighted F1** scores on the corresponding test set. *(15 points)*

(Hint: for each selected hyperparameter, you need to prepare a list of reasonable values for tuning. Use ***sklearn.model_selection.GridSearchCV*** *for grid search.)*

4) Besides accuracy and F1 scores, one can look at the confusion matrix to understand the model's prediction behaviour. Compare confusion matrix on test data before and after hyperparameter tuning for the selected model in 3). Discuss any of your observations or insights. *(5 points)*

(Hint: you can use ***confusion_matrix*** *from scikit-learn, and use* ***sb.heatmap(conf_matrix, annot=True)*** *to plot a corresponding heatmap for better visualization).*

5) You can get feature importances for some tree-based classifiers in scikit-learn via ***clf.feature_importances_***. Example code for plotting the feature importances for an adaboost decision tree classifier has been given in the IPython notebook. Generate the plot and describe any of your observations. Briefly describe how the feature importances for the given classifier can be computed. *(5 points)*.

6) Think about how to compute the feature importances for other classifiers in 1). Select one classifier (except decsion tree), and generate a similar bar plot showing the feature importances. *(8 points)*

4 Best Model Construction *(15 points)*

1) Generate the best classifier you can for predicting the diabetes outcome. Show the classification **accuracy**, **weighted F1**, and **confusion matrix** on the test data (you should keep the original train test split). Besides that, you can also show any result or plots that are helpful for us to better understand your model. Write a short description of your model indicating the elements that help to improve prediction. You may use any classifier including but not limited to those experimented above. Marks will be given based on your model performance and description. *(13 points)*

(Hint: you can think about improving the model performance from different aspects: e.g. improve the feature set; choose a more sophisticated classifier such as ensemble models; or improve the training process via careful hyperparameter tunning or sampling techniques.)

2) Can the final model you constructed be deployed at the hospital to diagnose patients automatically? How would the doctors explain the prediction results to the patients? Write down any of your opinions or concerns. *(2 points)*