Assignment 3: Cervical cancer screening prediction using Python

CS253 – 2023

IIT Kanpur

Classification refers to categorizing the given data into classes. For example,
- Given an image of hand-written character, identifying the character (multi-class classification)
- Given an image, annotating it with all the objects present in the image (multi-label classification)
- Classifying an email as spam or non-spam (binary classification)
- Classifying a tumor as benign or malignant and so on

In this assignment, you will be building a classifier for cervical cancer screening prediction. You will be using the Kaggle dataset [Cervical Cancer Risk Factors Dataset] (https://www.kaggle.com/datasets/loveall/cervical-cancer-risk-classification) for this task.

The data has 32 risk factors and 4 target variables. Target variables are different screening/diagnostic techniques namely Hinselmann, Schiller, Cytology and Biopsy.

You have to perform BINARY CLASSIFICATION for each of the 4 target variables individually.

Note: You will only use the libraries mentioned in this assignment.

DATA PREPROCESSING

- Deal with missing values: You can use either data elimination or data imputation to deal with missing data.
- Encode the categorical data (if needed).
- Identify and Remove the outliers.
- Normalize the data.
- Balance the data (in case of imbalanced classes).

FEATURE EXTRACTION

- Identify useful features and eliminate redundant features
- Reduce the dimensionality of data (optional, in case of high dimensional data)
- Use PCA to extract the principal components.

DATA BALANCING

- Use SMOTE or ADASYN to balance the classes.
  (SMOTE refers to Synthetic Minority Oversampling Technique which involves oversampling the minority class examples based on nearby feature space while ADASYN utilizes the density of minority class examples to generate synthetic samples).
  More details can be found at https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

DATA VISUALIZATION

- Visualize the normalized data distribution using boxplot.
- Identify correlated features using correlation heatmap.
- Plot the confusion matrix.
  You can use seaborn and matplotlib libraries for visualization.

CLASSIFIER

- You will use two classifiers for this task. SVM [https://scikit-learn.org/stable/modules/svm.html] and KNN [https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html]
  (You can experiment with different train test split ratios to obtain higher evaluation scores)

  Make sure to test your results by tuning the hyperparameters for each classifier, e.g., regularizer weight for soft-margin in SVM and the value of k in KNN. You can fine-tune these hyperparameters on the validation split and then report the results for the best value on the test dataset.

EVALUATION

- Three evaluation metrics will be used for each classifier: Accuracy, Precision and Recall. (Show them in a Tabular format considering all the cases)
- Plot the confusion matrix for each target variable.

Note: You have to perform classification for each of the target variables individually for each classifier.

Finally, Submit the code as a Jupyter Notebook.

Note: Any kind of plagiarism will be penalized.

Questions related to assignment can be directed to Saqib Sarwar (saqib@cse.iitk.ac.in)