CPSC 483 - Data Mining and Pattern Recognition

Fall 2017 - Project 2, due November 30

In this project, you will use scikit-learn and related Python libraries to perform classification and clustering experiments against data from the UCI Machine Learning Repository.

Choosing a dataset

UC Irvine maintains an archive of datasets for use in machine learning experiments. For each dataset, the repository lists “Associated Tasks” including regression, classification, and clustering.

Browse through the available datasets and choose one suitable for classification. Since the experiments below include clustering, you may wish to choose a dataset that is shown as suitable for both classification and clustering. Be careful, however, not to choose a dataset whose only listed task is clustering, since there will be no class labels.

In the data folder for each dataset you should find a `.names` file (or equivalent) describing the features. Depending on the dataset, you may also need to combine data files, select a subset of the data, or preprocess the data before it can be used. See Loading from external datasets for tips.

Allocating time for this project

Note that while the due date of this project isn’t until after Fall Recess, it should not take three weeks to complete. Do not get stuck on this project and neglect the Kaggle competition.

Working with other students

You may work alone on this project, or with students in your team for the group project. If you work alone, any code you submit must be your own. If you work with a team, any code submitted must be written by members of that team.

Experiments

1. Use `sklearn.model_selection.train_test_split` to split your data into training and test sets.

2. Fit an `sklearn.naive_bayes.GaussianNB` classifier to your dataset. (This is the classifier from Section 5.1.2.5 of the textbook; the classifier in Section 5.1.2.6 is `sklearn.naive_bayes.MultinomialNB`.)

3. Use your Naive Bayes classifier to predict the labels for your dataset, and evaluate its performance. If your dataset has multiple classes, use `sklearn.metrics.confusion_matrix`. For a binary classifier, plot an `sklearn.metrics.roc_curve`.

4. Fit an `sklearn.neighbors.KNeighborsClassifier` to your dataset (Section 5.3.1) and evaluate its `sklearn.metrics.zero_one_loss` over the range `n_neighbors` = 5 to `n_neighbors` = $\sqrt{N}$.

5. Compare the performance of the Naive Bayes classifier in experiment *(3)* to the KNN classifier with the best choice for *K* from experiment *(4)*.

6. Fit an `sklearn.svm.SVC` to your dataset (section 5.3.2). Compare the `sklearn.metrics.zero_one_loss` of `kernel='linear'` and `kernel='poly'` with `degree` = 3 and `degree` = 5.

7. Compare the performance of the best SVM from experiment *(6)* with the performance in experiments *(3)* and *(5)*. Which classification algorithm performed best for your dataset?

8. Remove the labels from your dataset and run `sklearn.cluster.KMeans` on the features, with `n_clusters` set to the number of classes in your dataset. How do the `labels_` found by the clustering algorithm compare to the original labels?

Documentation

In your Python code, use comments to clearly identify

- which experiment is being performed
- the output produced by each command
- the results of the comparisons in experiments 3-8

Submission

Turn in the Python code for your project by placing it in the `project2/` subdirectory of the folder that will be shared with you on Dropbox. If you work in a team, turn in only one submission.