Assignment 2– Introduction to sklearn for Machine Learning

Assignment overview. This assignment is designed to introduce you the sklearn library for machine learning in Python. This Assignment requires you to install sklearn and download the datasets **iris** and **20newsgroups text**, to learn and predict some items by using algorithms implemented in sklearn, evaluate model the models, convert text to vectors, and chain multiple estimators into one using a pipeline. Follow the steps as indicated and complete the tasks. You are expected to figure out details of syntax by consulting Python's Help function. Since the material builds on each other, it is recommended to follow the exercises in order. Please answer each question as needed by copying and commenting on the material that you produce in the IPython console as well as all scripts you are asked to create, or use Jupyter Notebook and download it as a Python file.

Submission. Create a folder called ML_Assignment2 and put all the files inside the folder. Compress this folder to create either ML_Assignment2.zip or ML_Assignment2.rar. Submit this compressed folder as your assignment submission on Brightspace.

Submission deadline. Thursday, 21 Sep, 10:00 pm.

Late submission policy: Submissions after the due date are penalized by a 10% grade reduction per day.

Academic Integrity: Dalhousie academic integrity policy applies to all submissions in this course. You are expected to submit your own work. Please refer to and understand the academic integrity policy, available at https://www.dal.ca/academicintegrity

Python: We will be using Python for the programing exercises based on scientific Python libraries like

- **numpy** - mainly useful for its *N*-dimensional array objects.
- **matplotlib** - 2D plotting library producing publication quality figures.
- **scikit-learn** - the machine learning algorithms used for data analysis and data mining tasks.

If you have question: Teaching Assistants (TAs) will be present during the labs to help you with any questions you may have. If you still have questions, feel free to email me at tt@cs.dal.ca.

Questions:

1. **[40 marks]** This Assignment requires you to write a Python script file called sol1.py.
   1.1. **[5 marks]** Import the ***iris*** dataset (`load_iris`) from *`sklearn datasets`* and place them in a variable `iris` (`iris` is a dot-accessible dictionary or a bunch object). Similar to the example in the course manuscript (section 2.4), apply a Linear SVM on the iris data and name the model `svc_iris`. Then build two models for sepal and petal features and name them as `svc_sepal` and `svc_petal` respectively. Finally, use these three models for predicting the labels for the classes and save them in `predicted_iris`, `predicted_sepal`, and `predicted_petal` arrays. **What** is the role of the `fit` and `predict` methods?
   1.2. **[25 marks, 20 marks for Grads]** Calculate and show the quantities *Accuracy*, *Precision*, *Recall*, and *F1-score* as well as the confusion matrix for all three models (hint: you might use `score` method in the sklearn.metrics module). **Compare** the results and **explain** why there are differences in the metric

values? (Hint: You might refer to the question 1.9 of assignment 1 and check with the confusion matrixes.)

1.3. **[10 marks]** Estimate the accuracy of a linear SVM on the iris dataset (all data, sepal, and petal) using 10-fold cross validation and show the *mean* and *standard deviation* of the accuracy (hint: sklearn provided you the `model_selection.cross_val_score` function). **Explain** briefly what k-fold cross validation is and what it is used for. **Repeat** the experiment with 5-fold cross validation and compare the results.

1.4. **[Required for Grads - 5 marks, Bonus points for UnderGrads - 5 marks]** Without using sklearn methods related to cross validation, write a function to implement cross validation and compare the results.

2. **[30 marks]** This Assignment requires you to write a Python script file called sol2.py.

2.1. **[5 marks]** Import the ***20 newsgroups text*** dataset (`fetch_20newsgroups`) from *`sklearn datasets`*. Read through the `fetch_20newsgroups` help and come to understand how you can load training and test sets with different categories. Use the `subset`, `shuffle`, and `random_state` parameters to load training and test data into bunch objects called `train_20news` and `test_20news`. **What** is the role of the `categories` parameter?

2.2. **[10 marks]** If you want to apply machine learning methods on text data, you should convert the text contents into the numerical feature vectors. The *`sklearn.feature_extraction.text`* sub-module has provided functions to convert text contents into feature vectors. Explore *`CountVectorizer`* and *`TfidfTransformer`* as well as *`TfidfVectorizer`* classes as well as *`fit`*, *`fit_transform`* and *`transform`* methods to figure out how to convert text contents into feature vectors. **Explain** briefly how these modules and methods perform the conversion. Convert the *training data* and the *test data* into feature vectors and place the results in `train_vectors` and `test_vectors`. Hint: for the `test_vectors`, you should call *`transform`* instead of *`fit_transform`*.

2.3. **[5 marks]** Similar to section 2.5 in the manuscript, apply a random forest classifier with 50 trees in the forest on the `train_vectors` and calculate and show *Accuracy, Precision*, *Recall*, and *F1-score* quantities as well as confusion matrix of the model on the `test_vectors`.

2.4. **[5 marks]** The `Pipeline` class is provided by sklearn to sequentially apply a list of transforms and a final estimator. Using the `Pipeline` class, apply a random forest classifier with 50 trees in the forest on the *training data*. Then calculate and show *Accuracy, Precision*, *Recall*, and *F1-score* quantities as well as confusion matrix of the model on the *test data*.

2.5. **[5 marks]** Repeat the 2.4 question using MLP classifier with 3 hidden layers with size 10, 20, and 10, and maximum iteration 10. Hint: use `sklearn.neural_network.MLPClassifier`.

3. **[30 marks]** This question requires you to write a Python script file called sol3.py. Please download the wine.zip file and extract it to the directory for this assignment. Read through the wine_names.txt file and come to understand the problem and the *wine* data contained in the wine.train dataset. Train one of the models **SVM**, **MLP**, or **RF** to develop the best possible model for classifying the *wine* data in the hold-out test data set of 58 records in the wine.test file given the training data. You must submit a list of 58 classifications (as a separate *.csv file) for the hold-out test set in the same order as received and we will use your answers to score how well your model performs. **Describe** your methodology for determining the best model. **Deploy** your best model by developing one final model using the best choice of learning parameters and all the training data. Everyone will be ranked based on how well they do on their classification of the hold-out test set and a **maximum 5 additional marks** will be given to each person based on their ranking.