EE769 Introduction to Machine Learning (March 2024)

Electrical Engineering, Indian Institute of Technology Bombay

Programming Assignment – 2 : Classification, Feature Engineering, Deployment

Instructions:

- Name files A2_<RollNo>.<extension>. Submit three files (or links to these): an .ipynb, and link to a video demo (approx 10 minutes; ensure that we can watch it, else we won't grade it)
- Use good coding practices such as avoiding hard-coding, using self-explanatory variable names, using functions (if applicable). This will also be graded.
- You may use libraries such as scikit-learn, and need not code anything from scratch.
- Cite your sources if you use code from the internet (line-by-line or block-by-block). Also clarify what you have modified.

1. Regression and out-of-distribution prediction:
    a. Download the wine quality datasets from https://archive.ics.uci.edu/ml/datasets/Wine+Quality
    b. Explore, visualize, and pre-process the data as appropriate. [1]
    c. Train, validate varying at least one hyperparameter, and test at least two types of models: [2]
        i. Random forest
        ii. Support vector regression with RBF kernel
        iii. Neural network with single hidden layer (output layer should have linear activation)
    d. Search the net about how to determine the importance of each variable, and find the importance in the final models tried. Comment on whether the same variables are important for different models. [1]
    e. Test the model for red with data from white and vice versa, and comment on whether the model for red wines is applicable to white wines and versa or not. [1]
2. Classification:
    a. Download the data to predict Down syndrome in mice from https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression#. The prediction problem is to either predict the genotype (binary) using the gene expression variables from DYRK1A_N to CaNA_N.
    b. Explore, visualize, and pre-process the data as appropriate, including developing a strategy to deal with missing variables. You can choose to impute the variable. The recommended way is to use multivariate feature imputation (https://scikit-learn.org/stable/modules/impute.html) [1]
    c. Train, validate varying at least one hyperparameter, and test at least two types of models: [2]
        i. Random forest
        ii. Support vector classification using RBF kernel
        iii. Neural network with single hidden layer (output layer should be have softmax activation)
    d. See if removing some features systematically will improve your models using recursive feature elimination https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html). [1]
3. Practice using pre-trained neural networks to extract domain-specific features for new tasks.
    a. Read the pytorch tutorial to use a pre-trained "ConvNet as fixed feature extractor" from https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html and you can ignore

"finetuning the ConvNet". Test this code out to see if it runs properly in your environment after eliminating code blocks that you do not need. [1]

   b. Write a function that outputs ResNet18 features for a given input image. Extract features for training images (in image_datasets['train']). You should get an Nx512 dimensional array. [1]
   c. Compare RBF kernel SVM (do grid search on kernel width and regularization) and random forest (do grid search on max depth and number of trees). Test the final model on test data and show the results -- accuracy and F1 score. [1]

4. Deploy one model from part 1 on a local webserver with a web frontend (e.g. using streamlit). Add some GUI elements, such as sliders for acidity, citrus etc. [2] Most of this should be in the video demo.