MIE 1624 Introduction to Data Science and Analytics – Winter 2018

Assignment 2

Due Date: 11:59 pm, April 6, 2018
Submit via Blackboard

Background:

Classification is a supervised machine learning approach required by several data mining tasks to classify data into classes. It is supervised because labelled data is used to train models. Many types of machine learning models can be used for training classifiers, such as the kNN algorithm, Logistic Regression, SVM, Decision Trees, XGBoost and Random Forest.

The purpose of this assignment is to train, validate, and tune multiple classifiers that can predict, given a set of observations about a person, which income bracket they fall into. The data for this assignment is census data from a previous year that contains information about people across the United States.

As seen in Assignment 1, data is often split into training and testing data. The training data is typically further divided to create validation sets, either by just splitting, if enough data exists, or by using **cross-validation** within the training set. The model can be iteratively improved by tuning the hyperparameters of the model or by feature selection.

Produce a report in the form of an IPython Notebook detailing the analysis you performed to determine the best classifier for the given data set. Your analysis must include the following steps: data cleaning, exploratory data analysis, feature selection (or model preparation), model implementation, model validation, model tuning, and discussion. When writing the report, make sure to explain for each step, what it is doing, why it is important, and the pros and cons of that approach.

The training set and testing set for this assignment are in the files *income-training.csv* and *income-testing.csv*, respectively. Both datasets are labelled, and the *IncomeBracket* column is the target variable, which has one of three possible values (`<50K`, `50-100K`, or `>100K`). The list of attributes contained in both datasets can be found in the appendix. The training set and testing set come from a modified version of the dataset found in the following UCI repository: https://archive.ics.uci.edu/ml/datasets/adult. More information can be found there, including papers that have used this data set.

Learning objectives:

1. Understand how to clean and prepare data for machine learning, including working with incomplete data, and categorical data.
2. Understand how to explore data to look for correlations between the features and the target variable.
3. Understand how to apply machine learning algorithms to the task of classification.
4. Improve on skills and competencies required to compare the performance of classification algorithms, including application of performance measurements, statistical hypothesis testing, and visualization of comparisons.
5. Understand how to improve the performance of your model.
6. Improve on skill and competencies required to collate and present domain specific, evidence-based insights.

To do:

The following sections should be included but the order does not need to be followed. The discussion for each section is included in that section's marks.

1. **Data cleaning (20 marks):**
   While the data is made ready for analysis, several values are missing, and majority of the features are categorical. For the data cleaning step, handle missing values however you see fit and justify your approach. Provide some insight on why you think the values are missing and how your approach might impact the overall analysis. Suggestions include filling the missing values with a certain value (e.g. mean for continuous data, mode for categorical data) and completely removing the features with missing values. Secondly, convert categorical data into numerical data by encoding and explain why you used this particular encoding method. These tasks can be done interchangeably i.e. encoding can be done first.

2. **Exploratory data analysis (15 marks):**
   a. Present 3 graphical figures that represent trends in the data. How could these trends be used to help with the task of classification of income bracket? All graphs should be ***readable*** and have all axes ***appropriately labelled***.
   b. Visualize the order of feature importance. Some possible methods include correlation plot, or a similar method. Given the data, which of the original attributes in the data are most related to an individual's income bracket?

   The steps specified before are not in a set order.

3. **Feature selection (10 marks):**
   Create at least one additional feature that is not originally part of the dataset but is based on the original features of the dataset. Explain how feature engineering is a useful tool in machine learning. Then select the features to be used for analysis either manually or through some feature selection algorithm (e.g. regularized regression). Not all features need to be used; features can be removed or added as desired although the same set of

features must be used for all your machine learning models. Provide justification on why you selected the set of features.

4. **Model implementation (25 marks):**
   Implement 4 different classification algorithms of your choice on the training data using 10-fold cross-validation. How does your model accuracy compare across the folds? What is average and variance of accuracy for folds? Which model performed best? Give the reason based on bias-variance trade-off. For each algorithm, briefly talk about what it does, what its pros and cons are, and why you chose that algorithm.

5. **Model tuning (20 marks):**
   Improve the performance of the models from the previous step and select a final optimal model using grid search
   (parameter sweep) based on a metric (or metrics) that you choose. Choosing an optimal model for a given task (comparing multiple classifiers on a specific domain) requires selecting performance measures, such as accuracy, true positive rate (TPR), false positive rate (FPR), etc, to compare the model performance. Explain how the chosen algorithm applies to the data. Regardless of your chosen performance measures, your optimal model must have an accuracy of at least 70% in cross-validation on the training set.

6. **Testing & Discussion (10 marks):**
   Use your optimal model to make predictions on the test set. How does your model perform on the test set vs. the training set? The overall fit of the model, how to increase the accuracy (test, training)? Is it overfitting or underfitting? Why?
   Insufficient discussion will lead to the deduction on marks.

Bonus:

We will give 10 bonus marks to 3 students who achieve the highest accuracy values on a different testing set than the one provided. A separate Python file must be created for evaluating the bonus section. In the separate file, you are asked to write a function called *bonus* that takes as input the paths of the training set and the testing set and outputs the predicted labels (e.g. `<50K`, `50-100K`, `>100K`) into a csv file. Inside the function, there should be a step that cleans and prepares the data for the model, a step that creates your optimal model from the assignment and trains it on the training data, and a step that evaluates the model on the testing data and produces the predicted labels.

This file should work independently so any required libraries should be imported. Grid search should not be implemented in the code, so the model should be trained using the optimal hyperparameters from the assignment. The bonus section will evaluate a different testing set than the one given with the assignment, but it will have the same structure. The naming conventions for the Python file and the csv output file are described in the *What to Submit* section below.

Here is a skeleton of the *bonus* function:

```
def bonus(training_file_path, testing_file_path):
   # code for cleaning the training data
   # code for training the model
   # code for predicting the labels of testing data
   # code for writing predicted labels in csv file
```

A csv file called *sample_bonus.csv* has been posted on Blackboard to show an example of what *bonus* should output. If your files cannot be imported, or if you fail to follow the above instructions, you will forfeit your opportunity to compete for bonus marks.

Tools:

- **Software:**
  - **Python Version 3.X** is required for this assignment. Your code should run on the Data Scientist Workbench (Kernel 3). All libraries are allowed but here is a list of the major libraries you might consider: Numpy, Scipy, Scikit, Matplotlib, Pandas.
  - No other tool or software besides Python **and its component libraries** can be used to touch the data files. For instance, using Microsoft Excel to clean the data is not allowed.

- **Required data files:**
  - **income-training.csv:** classified census data for several people across USA with their corresponding income bracket.
  - **income-testing.csv:** more classified data in the same structure as the income-training-set.csv.
  - The data files cannot be altered by any means. The IPython Notebooks will be run using local versions of these data files.

What to submit:

Submit via Blackboard an IPython notebook containing your implementation and motivation for all the steps of the analysis with the following naming convention:

lastname_studentnumber_assignment2.ipynb

If you wish to complete the bonus section, make sure to submit the following files as well:

lastname_studentnumber_assignment2_bonus.py

Make sure that you **comment** your code appropriately and describe each step in sufficient detail. Respect the above convention when naming your file, making sure that all letters are lowercase and underscores are used as shown. **A program that cannot be evaluated because it varies from specifications will receive zero marks. Late submissions will not be accepted.**

Tips:

1. You have a lot of freedom with however you want to approach each step and with whatever library or function you want to use. As open-ended as the problem seems, the emphasis of the assignment is for you to be able to explain the reasoning behind every step.

2. While some suggestions have been made in certain steps to give you some direction, you are not required to follow them. Doing them, however, guarantees full marks if implemented and explained correctly.

3. The output of the classifier when evaluated on the training set must be the same as the output of the classifier when evaluated on the testing set, but you may clean and prepare the data as you see fit for the training set and the testing set.

4. When evaluating the performance of two algorithms, keep in mind that there can be an inherent trade-off between the results on various performance measures. For example, the TPR and the FPR are quite different and often an algorithm with good results on one yields bad results on the other.

Appendix:

Below is the list of attributes/features contained in both the training set and testing set:

1. *Age:* Person's age (continuous)
2. *WorkClass:* Person's working status (Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked)
3. *FinalWeight*: Weight value generated by Current Population Survey (CPS) - people with similar demographic background should have similar weights within a given state (continuous)
4. *Education*: Highest education level completed by person (Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool)
5. *EducationLvl:* Numerical representation of the highest education level completed (discrete)
6. *MaritalStatus*: Person's marital status (Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse)
7. *Occupation*: Person's occupation (Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces)
8. *Relationship:* Relatives person has (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried)
9. *Race:* Person's race (White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black)
10. *Sex:* Peron's sex (Female, Male)
11. *CapitalGain*: Person's yearly capital gain on investments (continuous)
12. *CapitalLoss*: Person's yearly capital loss on investments (continuous)
13. *HoursPerWeek:* The number of hours the person works per week (continuous)
14. *NativeCountry:* Person's native country (United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Colombia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands)