MIE 1624 Introduction to Data Science and Analytics – Fall 2018

Assignment 2

Due Date: 11:59 pm, December 13th, 2018

Submit via Quercus

Background:

Kaggle is currently hosting an open data scientist competition titled "2018 Kaggle ML & DS Survey Challenge." The purpose of this challenge is to "tell a data story about a subset of the data science community represented in this survey, through a combination of both narrative text and data exploration." Kaggle is providing 6 monetary prizes for the best data storytelling submissions. More information on the competition, data, and prizes can be found on: https://www.kaggle.com/kaggle/kaggle-survey-2018

The dataset provided (**Kaggle_Salary.csv**) in Assignment 2 contains a modified version of the survey results provided by Kaggle in the file *mutiplechoiceResponses.csv*. The survey results from 15429 participants are shown in 395 columns, representing survey questions. Not all questions are answered by each participant, and responses contain various data types.

In the dataset for Assignment 2, Q9 "*What is your current yearly compensation (approximate $USD)?"* has been modified from a range to an integer to be used for regression. This has been done by replacing the compensation range with a random integer from a uniform distribution within that range. Rows with null values or undisclosed salaries have been dropped. For this assignment, only the file **Kaggle_Salary.csv** can be used. The column "Q9" contains the target variable, and an index column "index" has been added.

The purpose of this assignment is to
1) understand and explore employment in the data science community, as represented in a survey conducted by Kaggle.
2) train, validate, and tune multiple regressors that can predict, given a set of survey responses by a data scientist, what a survey respondent's current yearly compensation is.

Regression or **prediction** is a supervised machine learning approach used to predict a value of one variable when given the values of others. Many types of machine learning models can be used for training regressors, such as linear regression, decision trees, kNN, SVM, random forest, gradient-boosted decision trees and neural networks.

For the purposes of this assignment, any subset of **Kaggle_Salary.csv** can be used for data exploration and for regression purposes. For example, you may focus only on only one country, exclude features, or engineer new features. If a subset of data is chosen, **it must contain at least 5000 training points.**

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.

Submission:

1) Produce a report in the form of an IPython Notebook detailing the analysis you performed to determine the best regressor (prediction model) 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.

2) Create 5 slides in PowerPoint and PDF describing the findings from exploratory analysis, model feature importance, model results and visualizations.

Learning objectives:

1. Understand how to clean and prepare data for machine learning, including working with multiple data types, 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 regression/prediction.
4. Improve on skills and competencies required to compare the performance of prediction 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 some 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. 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 predicting yearly compensation or understanding the data? 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 a survey respondent's yearly compensation?

   The steps specified before are not in a set order.

3. **Feature selection (10 marks):**

   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. If the resulting number of features is very high, dimensionality reduction can also be used (e.g. PCA). Use at least one feature selection technique, and provide justification on why you selected the set of features.

4. **Model implementation (25 marks):**
   Implement 4 different regression/prediction 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 with hyperparameter tuning and select a final optimal model using grid search based on a metric (or metrics) that you choose. Choosing an optimal model for a given task (comparing multiple regressors on a specific domain) requires selecting performance measures, for example R^2 (coefficient of determination) and/or RMSE (root mean squared error) to compare the model performance. Explain how the chosen algorithm applies to the data.

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:

Implement a neural network to predict the target variable. Experiment with different neural network architectures (# of hidden layers, # number of nodes per layer) and parameters (learning rate, number of iterations, momentum).

We will give 10 bonus marks to up to 20 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 variable 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 values.

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 target of testing data
    # code for writing predicted target in csv file
```

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, Sklearn, 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:**
  - **Kaggle_Salary.csv**: survey responses with yearly compensation.
  - The data file cannot be altered by any means. The IPython Notebooks will be run using local version of this data file.

What to submit:

Submit via Quercus 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 regressor when evaluated on the training set must be the same as the output of the regressor 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 your algorithms, keep in mind that there can be an inherent trade-off between the results on various performance measures.