DAV 6150 Module 13 Assignment

Neural Networks

***** You may work in small groups of no more than three (3) people for this Assignment *****

The Module 4 Assignment made use of a data set (https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity) comprised of attributes that describe the characteristics of more than 39,600 online news articles. Please refer to the web page cited above for further details on these attributes.

As you will recall, for that Assignment you were tasked with apply feature selection and/or dimensionality reduction techniques to identify the explanatory variables to be included within a linear regression model that predicts the number of times an online news article will be shared. Your task for the **Module 13 Assignment** is to construct and compare / contrast the performance of three separate feed-forward / back propagating neural networks. The response variable you will be modeling will be **a categorical indicator variable** derived from the dataset's **share** attribute. Get started on the Assignment as follows:

1) Ensure the **M4_Data.csv** file (provided with the Module 4) has been loaded to your DAV 6150 Github Repository.
2) Then, using a Jupyter Notebook, read the data set from your Github repository and load it into a Pandas dataframe. Ensure your data attributes are properly labeled within the data frame.
3) Perform EDA work as necessary. (NOTE: If you already have a high-quality EDA from the M4 Assignment, you may incorporate it here. If your M4 Assignment EDA was flawed, you should repeat the EDA work and address any shortfalls identified in your M4 Assignment EDA. Note that any uncorrected flaws will result in corresponding point deductions for the M13 Assignment).
4) As the first step of your Data Preparation work, you ***MUST*** create a new categorical indicator variable derived from the content of the **share** attribute. As you will recall, the **share** attribute is a numeric representation of the number of times a given online news article has been shared. Using the results of your EDA, create a new indicator variable named "**share_level**" having the following three possible categorizations:

   A. "**low**": indicates that the number of shares for a given articles is less than ½ of the median number of shares for all news articles;

   B. "**medium**": indicates that the number of shares for a given articles is between 0.5 * the median number of shares for all news articles and 1.5 * the median number of shares for all news articles, i.e., (0.5 * median) < number of shares for the articles <= (1.5 * median)

   C. "**high**": indicates that the number of shares for a given article exceeds 1.5 * the median number of shares for all news articles.

   Ensure that an appropriate "**share_level**" value is calculated for every observation contained within the data set.

   Once you have created the "**share_level**" indicator, be sure to **remove the "share" attribute from your dataframe**. This must be done to eliminate the collinearity that will result from the addition of the "**share_level**" indicator to your collection of attributes.

5) Within your Prepped Data Review, be sure to analyze the distribution of the newly created "**share_level**" indicator value. What does your analysis tells us about the distribution of this newly created indicator variable?

6) Using your Python skills, apply your knowledge of feature selection and dimensionality reduction to the explanatory variables to identify variables that you believe will prove to be relatively useful within your models. Your work here should reflect some of the knowledge you have gained via your EDA work. While selecting your features, be sure to consider the tradeoff between model performance and model simplification, e.g., if you are reducing the complexity of your model, are you sacrificing too much in the way of accuracy (or some other performance measure)? The ways in which you implement your feature selection and/or dimensionality reduction decisions are up to you as a data science practitioner to determine: will you use filtering methods? PCA? Stepwise search? etc. It is up to you to decide upon your own preferred approach. Be sure to include an explanatory narrative that justifies your decision making process.

7) After splitting the data into training and testing subsets, use the training subset to construct and train at least three different Python-based feed-forward, back propagating neural network models using the same explanatory variables for each model. **Your models must each include at least four (4) explanatory variables.** The response variable for your models is the newly created **share_level** indicator attribute. Since your models will each be utilizing the same set of explanatory variables you should differentiate them via the implementation of varying hyperparameters, e.g., the number of layers, the number of neurons per layer, the learning rate, the type of activation function, etc. It is up to you as the data science practitioner to decide upon the Python tools to make use of for these tasks (the examples provided in the HOML text might be a good place to start). Be advised that the training of a neural network can at times be a very time consuming and resource-intensive process, so be sure to manage your computing resources accordingly.

8) After training your various models, decide how you will select the "best" neural network model from those you have constructed. For example, are you willing to select a model with slightly lower performance if it is easier to interpret or less complicated to implement? What metrics will you use to compare/contrast your models? Evaluate the performance of your models via cross validation using the training data set. Then apply your preferred model to the testing subset and assess how well it performs on that previously unseen data.

Your deliverable for this Assignment is your Jupyter Notebook. It should contain a combination of Python code cells and explanatory narratives contained within properly formatted Markdown cells. The Notebook should contain (at a minimum) the following sections (including the relevant Python code for each section):

1) **Introduction (5 Points)**: Summarize the problem + explain the steps you plan to take to address the problem

2) **Exploratory Data Analysis (10 Points)**: Explain + present your EDA work including any conclusions you draw from your analysis, including any preliminary predictive inferences. This section should include any Python code used for the EDA.

3) **Data Preparation (10 Points)**: Describe + show the steps you have taken to address the data integrity + usability issues you identified in your EDA, including any feature engineering techniques you have applied to the data set. This section should include any Python code used for Data Preparation.

4) **Prepped Data Review (5 Points)**: Explain + present your post-Data Prep EDA analysis. This section should include any Python code used for re-running your EDA on the variables adjusted during your Data Preparation work.

5) **Neural Network Modeling (45 Points)**: Explain + present your neural network modeling work, including your feature selection / dimensionality reduction decisions and the process by which you selected the hyperparameters for your models. This section should include any Python code used for feature selection, dimensionality reduction, and model building.

6) **Select Models (15 Points)**: Explain your model selection criteria. Identify your preferred model. Compare / contrast its performance with that of your other models. Discuss why you've selected that specific model as your preferred model. Apply your preferred model to the testing subset and discuss your results. Did your preferred model perform as well as expected? Be sure include any Python code used as part of your model selection work and to frame your discussion within the context of the classification performance metrics you have derived from the models.

7) **Conclusions (10 Points)**

Your Jupyter Notebook deliverable should be similar to that of a publication-quality / professional caliber document and should include clearly labeled graphics, high-quality formatting, clearly defined section and sub-section headers, and be free of spelling and grammar errors. Furthermore, your Pythion code should include succinct explanatory comments.

Upload your Jupyter Notebook within the provided M13 Assignment Canvas submission portal. Be sure to save your Notebook using the following nomenclature: **first initial_last name_M13_assn**" (e.g., J_Smith_M13_assn). ***Small groups should identity all group members at the start of the Jupyter Notebook and each team member should submit their own copy of the team's work within Canvas.***