BITS F464 – Machine Learning (2019-2020 Sem II)

Assignment 2 (Due Date – 21/04/2020)

General Instructions:

- Your report, code, and visualizations must be uploaded to CMS by any group member, and will be checked for plagiarism. The report should describe your design decisions, results, and any conclusions that can be drawn (e.g., why an algorithm performed good/bad).
- The algorithms must be written in C++/Java/Python. High level libraries (e.g., Scikit-learn, Gensim, etc.) may NOT be used. Data manipulation libraries (like NumPy and Pandas) are allowed (and recommended). For visualizations, any library can be used.
- Try using vector operations (like NumPy matrix multiplications) whenever possible to improve your implementation's scalability. **This is recommended but not mandatory**.
- Please email Vamsi Aribandi (f20160803@hyderabad.bits-pilani.ac.in) for any queries.

1) Logistic Regression:

- Use this dataset to evaluate your model. The aim is to detect forged banknotes.
- Use an 80-20 train-test split to train/evaluate your model, and report the accuracy and F-score for the test set. Train a model (i) without regularization, (ii) with L1 regularization, and (iii) with L2 regularization.
- Experiment with different learning rates and weight initializations (Gaussian, uniform, etc.).
- Try and to interpret any one of your trained models to determine feature importance. For example, a person's height is a much more relevant feature to predict their age than, say, their skin colour, and this can probably be shown (hint: what does a feature's weight in the model say about it?). Describe your approach and results in your report (references to credible blogs and research papers are recommended to strengthen your argument and justify your approach).
- It is suggested that you scale your features before training/interpreting the model so that they share the same mean, variance, and/or bounds. Mention how you scale them in your report.

2) Neural Networks:

- Use this dataset to evaluate your model. The aim is to predict whether a house's price will be above or below the market's median price (binary classification).
- Use an 80-20 train-test split to train/evaluate your model, and report the accuracy and F-score for the test set. Try to experiment with different configurations of the number of hidden layers, number of hidden neurons, activation functions, learning rates, and weight initializations (Gaussian, uniform, etc.).
- Plot a graph illustrating how the loss function's value decreases as training proceeds.
- *OPTIONAL*: improve your vanilla implementation by implementing Dropout, Batch Normalization, etc. and mention it in your report.

Suggested Reading:

- Bishop's Ch. 4.3, 5.1-5.3.