SC4001/4042: Neural Networks and Deep Learning

Programming Assignment

Part A: Classification problem

- DNN to classify the **Musical Genre** dataset: **~2000** audio tracks, spanning 3 seconds each.
- The dataset has been pre-processed and **57 features** have been extracted: **audio_gtzan.csv**.
- Classification task: classify whether the audio track (in the form of engineered features) belongs to the **blues** or **metal** genre.
- Start with **PartA_Template.ipynb**.

Part A

1. DNN with three hidden layer (128 ReLU units), GD with ‘Adam’ optimizer. Dropout of probability 0.2. Divide the dataset into 70:30 train and test. Use early-stopping.
2. Use 5-fold CV to determine the **optimal batch size** from {32, 64, 128, 256}. Report time-taken.

Part A

3. Use 5-fold CV to determine the **optimal number of hidden-layer** neurons from {64, 128, 256}.
4. Run model inference using the provided audio track data named ‘audio_test.wav’ (use the preprocessing function 'extract_features' in **common_utils.py**) and find **the most important features via SHAP**.

Part B: Regression problem

- The aim is to predict public housing prices in Singapore from related features (#10):
  - **Numeric features**: dist_to_nearest_stn, dist_to_dhoby, degree_centrality, eigenvector_centrality, remaining_lease_years, floor_area_sqm
  - **Categorical features**: month, town, flat_model_type, storey_range
- Data: **hdb_price_prediction.csv**.
- Several libraries to be used: Pytorch-Tabular (B1), Pytorch-WideDeep (B2), Captum (B3), Alibi Detect (B4).
- Start with **PartB_Template.ipynb**.

Part B1: modelling tabular data

1. Feedforward neural network with 1 hidden layer containing 50 neurons.
2. Divide the dataset into Train data: up to year 2020 (inclusive); Test data: for year 2021.
3. Use *DataConfig, TrainerConfig, CategoryEmbeddingModelConfig, OptimizerConfig*, and *TabularModel* from the **Pytorch-Tabular** library to define your data and create the final model.
4. Report evaluation metrics on test data.
5. Print out cases with the largest errors.

Part B2: modelling tabular data

1. Feedforward neural network with 2 hidden layers containing 200 and 100 neurons respectively.
2. Divide the dataset into Train data: year 2020 and before; Test data: year 2021 and after.
3. Use *TabProcessor, TabMlp,* and *Trainer* from the **Pytorch-WideDeep** library to define your data and final model for training.
4. Report evaluation metrics on test data.

Part B3: model explainability

1. Build a model using only **numeric** features.
2. Generate saliency scores via several model **XAI** explainability algorithms
   (Input x Gradients, Integrated Gradients, DeepLift, GradientSHAP, Feature Ablation), with the help of the library **Captum**.
3. Understand various model explainability algorithms, and analyze the three most important features for regression.

Part B4: model drift

1. Study whether model performance degrades on new data points.
    a. Test the model with 2022 data
    b. Test the model with 2023 data
2. Ways to categorise, quantify and detect data distribution shifts.
    a. Use the **Alibi Detect** library to perform appropriate statistical tests depending on the type of feature
3. Which features contribute to this shift?
4. Think of a simple way to address model degradation and try it out.

Notes

- Marking based on your codes in Jupyter notebooks.
- Marks: **45** for Part A + **45** for Part B + **10** for presentation.
- Late submissions: penalized for 5 marks for each day up to 3 days.
- This assignment is to be done **individually**. Absolutely NO copying, duplicating, or plagiarism. You can discuss it with your classmates, but your submission must be your own unique work.
- **Follow the format** in the 2 notebooks provided.
- Post your queries on the **Discussion Board** in NTULearn (TAs will update a list of FAQ in there).
- Approach TAs Xia Jing, Yan Yige, Tiara Natasha, and Ajith Senthisenan for help via deeplearning4001@gmail.com