SC4001/4042: Neural Networks and Deep Learning

Programming Assignment

Part A: Classification problem

- DNN to classify the National Speech Corpus dataset: ~12,057 data samples, spanning 3 - 30 seconds each.
- The dataset has been pre-processed and 77 features has been extracted: **simplified.csv**.
- Polarity detection task: classify whether the speech recording (in the form of engineered features) has positive or negative emotions.
- Begin with **PartAIntro.ipynb.**

Part A

1. DNN with three hidden layer (128 ReLU units), GD with ‘Adam’ optimizer. Dropout at p = 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 {128, 256, 512, 1024}. 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 record data named ‘record.wav’ (use the preprocessing code provided in **data_preprocess.ipynb**) 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)

Part B1: modelling tabular data

Start with **PartB_1.ipynb.**

1. Feedforward neural network with 1 hidden layer containing 50 neurons.
2. Divide the dataset into Train data: up to year 2019 (inclusive); Validation data: year 2020; 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. Analyse cases with the largest errors

Part B2: modelling tabular data

Start with **PartB_2.ipynb.**

1. Feedforward neural network with 2 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

Start with **PartB_3.ipynb.**

1. Build a model using only **numeric** features
2. Generate saliency scores via several model explainability algos (Saliency, Input x Gradients, Integrated Gradients, GradientSHAP, Feature Ablation), with the help of the library **Captum**
3. Understand the importance of the choice of **baselines** by examining what happens when features are normalised

Part B4: model drift

Start with **PartB_4.ipynb.**

1. Study whether model performance degrades on new data points
    a. Policy changes → possible changes in data distribution
    b. Load the saved model from Part B1 when performing the evaluation
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. Think of a simple way to address model degradation and try it out
4. Which features are shifting with cooling measures?

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 specified format** in the 8 notebooks provided.
- Post your queries on the **Discussion Board** in NTULearn
  (TAs will update a list of FAQ in there).
- Approach TAs Charlene, Yihao, Shreyas, Xia Jing for help via
  neuralnetworks4042@gmail.com