Statistical Learning: Final Project

Deadline: 11 February 2025 at 23:59

Instructions

Dataset Selection

Select a dataset that interests you from the options below:

- Apple Quality
- House Rent Data
- American Sign Language MNIST
- Spam Emails
- European Soccer Data

Task 1: Build a Deep Feedforward Neural Network (30%)

Use **Keras/Torch** to construct **a deep feedforward neural network** (excluding convolutional and recurrent layers) with a minimum of **three hidden layers** to predict either a **category** (classification) or a **continuous value** (regression).

Model Requirements

Ensure that your neural network:

- Has properly **cleaned** data (e.g., one-hot encoding, removal of symbols like dollar signs, handling missing values).
- Uses **scaled or normalized** data (e.g., Z-score standardization) before training.
- Has a **suitable architecture and loss function** for the problem.
- Reports at least **two evaluation metrics** for both training and test data.

Task 2: Baseline Comparison with Traditional SL Models (30%)

To determine whether a neural network was necessary, build a simpler **statistical learning model** from this course and compare the performance. Make sure to **use the same predictors and outcome**. Possible models include:

- Linear Regression
- Logistic Regression
- KNN Regression
- KNN Classifier
- Decision Tree Regression
- Decision Tree Classifier
- Random Forest Classifier
- Random Forest Regression

Choose **one** of these models and compare the performance against your neural network to justify and convince me whether deep learning was necessary for the task or not.

Note: If you fail at building a neural network, you can still present and evaluate the performance of the other chosen model you built.

Task 3: Write Techincal Report (40%)

Lastly, create a technical report of **maximum 5 pages** discussing your model building process, the results, and your reflection on it. The report should follow the format in the example including an Introduction, Analysis, Methods, Results, and Reflection section. Your report is practice for presenting results to non-technical audiences in your Data Science career.

Technical Report Sections

Your final submission should include a **technical report** following the structure given below (`.qmd`-template available on Ilias):

Introduction

Introduce the problem and dataset, provide background information, and explain its importance.

Analysis

Summarize the dataset with exploratory data analysis, including summary statistics, correlation heatmaps, and visualizations. Discuss feature selection and data preprocessing.

Methods

Describe your model architecture and training approach. Explain changes made during the process in a way that a non-technical audience can understand.

Results

Provide a detailed comparison of model performance, including whether deep learning was necessary.

Reflection

Discuss lessons learned, challenges faced, and potential improvements for the future.

What to Submit

1. **PDF of your technical report** that is **maximum 5 pages** including tables and figures (rendered through Quarto, you must use the template provided). Remember, quality is more important than quantity!
2. **Code submission** as a rendered `.html` file **with resources embedded** via Quarto or a **GitHub link** with timestamped edits. Provide short comments of what you are doing before each chunk of code.

Marking Criteria

You will be judged on the following rubric criteria:

- Quality of technical report
  - Introduction provides reader with all the necessary background and information about the data, and importance of the problem being solved is apparent
  - Analysis demonstrates a deep understanding of the data and provides the reader with the context necessary to understand both the problem and the modelling choices.

  - Methods is complete, appropriate for the task, and would allow someone to roughly recreate the model(s) used. Reasoning behind choices made when building the model are provided.
  - Results provides a nuanced and detailed discussion of the model performance (e.g. beyond simple overall metrics), and discusses the implication of these results (e.g. how they affect whether the model could be used, whether performance differs for different groups, etc.)
  - Reflection demonstrates that student(s) understand the problem at hand, had an organized way of approaching the problem, recognize limitations of the curent analysis, and have thought about improvements taht can be made to the model(s) in the future.

- Code is organized and appropriate for task (e.g. the architecture chosen is appropriate for the problem, the loss function matches the desired output, metrics are appropriate for the task, no data leakage, code runs as intended…etc.)

- Quality of problem (the problem being solved and the questions being answered are sufficiently complex)