ICS 435 Final Project Requirements

Project

The ICS 435 Project is a chance to apply ML models to a problem that interests you. You encouraged to use common tools such as scikit-learn, Keras+Tensorflow, or Pytorch. Remember you can turn any tabular data matrix into a machine learning problem by trying to predict one variable from another set of variables. Sources of data and problems.:

- UCI ML Repository of ML datasets. This is the easiest, because most of the datasets are already processed and relatively easy to use. https://archive.ics.uci.edu/ml/index.php
- Kaggle. These often come with example code of how to download and process the data. https://www.kaggle.com
- Dataset Search https://datasetsearch.research.google.com/

If you have other ideas for a project, such as building a new dataset (for example, a Pokemon-MNIST benchmark dataset would get a lot of citations) or implementing a complicated neural network from scratch, feel free to email the instructor.

Grading Rubric

Implementation: Code and Analysis (10 points)

- Creativity.
- The code and analysis are correct.
- The code is well-organized.
- Model performance has been evaluated appropriately.

Write-Up (10 points):

1. **Problem specification** (2 points) Describe the problem and why a data-driven machine learning is an appropriate approach.
2. **Model specification** (1 points) What model are you using? What are the inputs and outputs? What is the loss function?
3. **Specify features and pre-processing** (1 point) Describe the feature representation and any transformations applied. This may include:
    - Units and feature types (E.G. categorical, ordinal, real, one-hot).
    - Transformations (E.G. min-max scaling, standardization, $\log(x+1)$).

- Missing data (E.G. Removal, Padding, Interpolation, Imputation)
- Data augmentation (E.G. Gaussian noise, cropping, translation).

4. **Specify data split strategy** (1 point) Describe data splits and how they are used, or how cross-validation was used.

   Example: *The total dataset contained 1,000,000 examples. The data was randomly permuted, then divided into three subsets: 60% training, 20% validation, and 20% test. Models were trained on the training set, while the validation set was used for early stopping, hyperparameter optimization, and model selection. The test set was used to evaluate the final model.*

5. **Specify the hyperparameter search space** (1 point) This is the list of hyperparameters that were optimized, and the range of values that were explored. This can be difficult to describe succinctly since hyperparameter tuning is usually an iterative process involving the experimenter. Ideally, you use a systematic hyperparameter optimization framework like sklearn's GridSearchCV, but if you do the hyperparameter optimization manually, you can simply state the range of values you explored (min and max) for each hyperparameter and the total number of models you tried.

   Example: *For the K-Nearest Neighbor classifier we tried different values of K and different distance metrics. We tried all odd values of K from the set of integers between 1 and 99, {1,3,5,...,99}. We tried the L1 and L2 distance metrics.*

6. **Explain how hyperparameters were optimized** (1 point) If you tuned the hyperparameters by hand, or exhaustively tested every hyperparameter combination in the search space, this can be a simple statement of the metric and validation set used. State any optimization algorithms used, e.g. Random Search, Grid Search, etc.

   Example: *After trying all combinations of hyperparameters in the search space, the model with the highest accuracy on the validation set was selected.*

7. **Evaluate model on clean test set** (1 point) When quantifying performance, remember to specify the *metric* and *dataset* for every number you present. It is common to compare multiple models on a test set, which technically means the test set is no longer perfectly clean — this is OK, as long as you do not compare a *large* number of models or hyperparameter combinations on the test set.

8. **Explain any differences in the train/test datasets** (1 point) Describe the tasks that you expect your model to generalize to, and reasons that the performance might decrease due to dataset shift.

   Example: *The test set is from a later time than the training set, so data drift could harm model performance.*