Final Project – Machine Learning (CS4342, Whitehill, Spring 2023)

1 Teams

For the final homework assignment, you should form teams of at least 3 people and at most 4 people. Each team will spend the last few weeks of the course tackling a particular Kaggle competition. You are free to choose any competition that is **not** a "Getting Started" competition or a "Limited-Participation" competition. It is ok, however, if the competition has already closed.

2 Learning Goals

At a high level, your learning goals for this project, and your chosen competition, are to: (1) Apply the theory you have been learning in class to tackle a real-world problem. (2) Practice using some off-the-shelf machine learning software. In terms of machine learning practice, your first step will be to understand what you are trying to predict/perceive (the y values), and what kinds of features you have to predict/perceive them from (the x values). In terms of software, you will likely start with sklearn for the shallow models because it offers a uniform interface to experiment with different models and has a relatively shallow learning curve. For the deep models you apply, you should try more specialized packages, e.g. Keras for neural network training. Here is a link to help you get started: `https://www.tensorflow.org/tutorials/keras/classification`

3 Points

You will earn points by successfully completing certain milestones:

- **Competition Selection and Computation of Baseline Accuracy/Loss (10 points)**: Choose a Kaggle competition that is **not** a "Getting Started" competition or a "Limited-Participation" competition. It does not have to be an active one (i.e., you do not need to be able to submit your solution to the leaderboard), but you do need to be able to access the data. In particular, there needs to be a set of test examples and associated test labels so that you can evaluate how good your model is. Pick a reasonable baseline method, e.g., random guessing, picking the mean training y value (for regression problems) or the most frequent y label (for classification problems). For the accuracy or loss function dictated by the competition, how well does this method do on the test data? This needs to be clearly stated in your report.

- **Visualize the Data (5 points)**: Use principal component analysis (PCA) to visualize (a subset of) the training data and the associated labels (use different colors to represent the y values). Comment briefly in your report on how separable, or explainable, the y values are based on the "raw" features x. Note that, if the data are linearly separable in PCA space, then the data are certainly linearly separable in the raw feature space; however, the converse is *not* true.

- **Shallow Model (10 points)**: Explore a shallow (e.g., linear or softmax regression, SVMs, random forests, boosting models, etc.) model to tackle your selected Kaggle competition. How well do you expect your model to perform on test data that were never seen during training, and never used for hyperparameter optimization? For your report, you need to describe your procedures concisely but precisely and use them to justify your testing accuracy/loss estimate.

- **3-layer Neural Network (10 points)**: Apply a 3-layer (i.e., neural network exactly 1 hidden layer) to your Kaggle competition. You can use your homework 5 code for this (which will work for just 1 hidden layer). Alternatively, you may use Keras, TensorFlow, PyTorch, etc.

- **Deep Neural Network (10 points)**: Use Keras, TensorFlow, PyTorch, etc. to systematically try deeper (> 3 layers) neural networks and see if this results in a significant accuracy gain. Make sure that you optimize your architecture (number of layers, number of neurons per layer, convolutional versus fully connected, etc.) only on training/validation data! In your report, you will need to describe your procedures in detail.
- **Creative and Technically Sound Machine Learning (20 points)**: What are the methodologically sound ways of improving the generalization accuracy of your model? Might feature engineering – i.e., based on intuition or some background research, devising novel functions of the "raw" features that produce new features that might better model the data – be useful? Could you harness pre-trained neural networks for related tasks and fine-tune them to your competition dataset? How much more oomph can you get out of your models by thorough and careful hyperparameter optimization? Might there be ways of augmenting your training and validation datasets through label-preserving transformations or self-supervision? The primary metric for this item is **the amount of demonstrated work and ingenuity towards tackling your problem**. A "reasonably good" job here will earn about 10 points; a "very nice job" will earn about 15 points; and a "truly exceptional" job can earn 20 points.
- **Final Report & Presentation (5 points)**: A precise and concise 2-page report, along with a very simple 2-slide presentation on Google Slides. You will get full credit on this item as long as you produce a report and give a short (1-2 minute) presentation. **Important**: even though this item is worth very few points, **it is the basis by which I can tell what you did for all the other items and thus how I will grade your project**. Hence, even if your report isn't elegant, **make sure that you convey all the cool stuff and hard work you did** for the project.

4 Deliverables

There are two deliverables:

- **Models: Implementation and Training Code**: A sequence of machine learning models, and associated training procedures, to tackle your chosen prediction problem. The code (typically in Python, though it can be in any programming language of your choice) necessary to train your models will be submitted in a Zip file. Most teams will use off-the-shelf software (e.g., Keras, sklearn), in which case the implementation is given to you. However, some teams might possibly implement their own models (e.g., a variant of the stepwise classification from homework), in which case their implementations should be submitted as well.
- **Report**: A simple report that describes precisely and concisely. **Important**: Make sure that if you borrow an idea from another researcher that you cite them (names, title of their work, year, and where it was published). The format of your report should be as follows:
    1. Title of the competition, list of students in your group and your email addresses.
    2. Introduction: succinctly but precisely describe what the goal of the competition is, including (a) what you are trying to predict/perceive; (b) what you have to predict/perceive with; (c) why the problem is important or interesting.
    3. Methods: this is the bulk of your report. Describe the models and techniques that you tried, starting from a **very** simple baseline model. For each model, report its performance, what you learned from the experiment, and which model/technique you therefore decided to try next. The shallow models you try will likely come first, and the deep ones will come later.
    4. Table of Results: Include a table that lists succinctly all the models/techniques you tried and their associated accuracy/loss values.
    5. Conclusions: For specific techniques that you tried (e.g., feature engineering according to a specific strategy), did it help? Which worked better – the shallow or the deep models?
    6. References: Did you borrow ideas from anyone else's work? If so, you must cite them!

5 Submission

Submit your report and code in a file called `homework6_WPIUSERNAME1_WPIUSERNAME2....zip`.