Your name:

B565 HW6 (Fall 2023)

Submit a PDF file to canvas. In addition, submit your code/notebook to github.iu under HW6 folder in your B565 repository.

1. (30 points) Linear regression
   (a) Dataset: the auto-mpg dataset on Kaggle.
   (b) (10 points) Perform feature scaling (using L2 norm) over the auto data set. Use two thirds of the data for training and the remaining one third for testing. Train a multivariate linear regression (`sklearn.linear_model.LinearRegression`) with "mpg" as the response and all other variables except "car name" as the predictors. What's the coefficient for the "year" attribute, and what does the coefficient suggest? What's the accuracy (mean squared error) of the model on the test data (one third of the mpg data set)?
   (c) (10 points) Try linear regression with regularization (Ridge and Lasso) as implemented in sklearn (RidgeCV and LassoCV). Use the cross-validation approach and compare the coefficients for the different attributes.
   (d) (10 points) Finally, compare the results obtained for ordinary linear regression, Ridge, and Lasso (using the α values that gave the lowest test MSE for the latter two). Does the type of regularization used affect the importance of the attributes? How can you interpret these results?

2. (30 points) Decision trees and ensemble approaches.
   - Use sklearn's breast cancer data set (from sklearn.datasets import load_breast_cancer)
   - Try the bagging and adaboost approaches using the decision tree as the base predictor. Experiment different parameters (e.g., number of base predictors). You may use BaggingClassifier and AdaBoostClassifier in sklearn.ensemble for this problem.
   - Document what you have tried and report your results.

3. (40 points) Using ANN and CNN.
   - Read about this tutorial on Tensorflow and examine the provided code for classifying images of clothing (keras.datasets.fashion_mnist) using an ANN.
   - How many neurons does the hidden layer have in the given ANN?
   - Try different numbers of neurons and report how the results change. Also try dropout (with different values) and report its impacts on the performance of the model. You may run the code in google colab and experiment with different settings there.
   - Try to implement a CNN based on the given ANN, by adding two convolution layers before the fully connected hidden layer and test different settings (e.g., number/size of filters). Summarize the experiments you have tried and results you get.

- Prepare learning curve plots (loss vs epoch, and accuracy vs epoch) for both ANN and CNN, and write a brief a summary of what you learn from the learning curves.
- Here are some hints about using CNN:

  ```
  #import Conv2D
  from keras.layers import Conv2D
  # create a model
  model = keras.Sequential()
  # add a convolution layer with 32 filters of 3 × 3
  model.add(keras.layers.Conv2D(filters=32, kernel_size=(3, 3), ⋯ )
  #Include additional parameters, input_shape = (28, 28, 1), data_format=“channels_last”)
  in Conv2D().
  ```

- Because the images used in this example (fashion_mnist) are in gray scales (of 28 × 28 pixels), the image data needs to be reformatted to be used as input to the convolution layer, e.g.,

  ```
  train_images = tf.reshape(train_images, shape=[-1, 28, 28, 1])
  ```

- Final note: some students may have some experience with PyTorch and would prefer PyTorch over TensorFlow. However, for this assignment, please stick to TensorFlow, just for consistency.