ECE 563 - Spring 2024

AI in Smart Grid

Project 1 - Due 6 Feb 2024

In this project, we will use the Advertising.csv dataset from our Gradient Descent example, the Scikit-Learn Breast Cancer dataset, and the Scikit-Learn Diabetes dataset.

We will explore several of the fundamental supervised learning algorithms, including Linear Regression, Logistic Regression, Decision Trees, Support Vector Machines, and k-Nearest Neighbors.

Deliverables

1. A brief overview of the project.
2. Python code that uses the Scikit-Learn Linear Regression algorithm to predict the number of unit sales given an amount of money spent on radio advertising. You can extract the radio advertising data from the Advertising CSV file as follows (there are other approaches, so do what works for you):

   ```
   df = pd.read_csv("Advertising.csv")
   x_arr = df["radio"].to_numpy()
   y_arr = df["sales"].to_numpy()
   x_arr = x_arr.reshape(-1,1)
   ```

   Since we are only interested in the best fit curve, you may use all the data for training.
3. Outputs: State the linear model coefficients determined by Linear Regression and the estimated # of units sold given a radio advertising budget of 23 M$ (units used for the radio advertising data).
4. Observations: How well do your coefficients match the results from our Gradient Descent example? Which method is faster for training the model?
5. Python code that uses the Scikit-Learn Logistic Regression algorithm to predict the presence of cancer based on the processed medical image data in the breast cancer dataset. For this part of the project, we will randomly select a subset of features for our classification problem. Use the following code to adjust the seed for the pseudo-random number generator that will randomly select 4 features:

   ```
   seed = 123456
   rng = np.random.default_rng(seed)
   idx_feat = (np.floor(30*rng.uniform(size=4))).astype(int)
   X = cancer["data"][:,idx_feat]
   ```

   Again, you may use the entire set of examples in the dataset for training the model. There is no need to set aside a test set for this project.
6. Outputs: State the seed value, the feature names, and the logistic regression score for the two best and two worst combinations of features found during your testing. Don't worry about finding the optimal solution. We are only interested in gaining some familiarity with the variation in the accuracty of the model.
7. Questions: As you varied the "seed" value, what changes did you notice in the random selection of features? Were any of the features better in terms of increasing the logistic regression score? If you increased the number of features selected, would you get higher scores? What tradeoff would you be making if you selected more features? What other hyperparameters could you tune to improve

the scores?

8. Python code that uses the Scikit-Learn Decision Tree Regressor algorithm to predict diabetes progression based on a subset of medical features. Similar to the Logistic Regression task above, use the pseudo-random number generator and "seed" to randomly select 4 features for building a decision tree. Use the entire set of examples in the dataset for training the model.
9. Outputs: State the seed value, the feature names, the depth of the tree, the number of leaves of the tree and the decision tree coefficient of determination score for the two best and two worst combinations of features found during your testing. Don't worry about finding the optimal solution.
10. Questions: As you varied the "seed" value, what changes did you notice in the random selection of features? Were any of the features better in terms of increasing the decision tree score? If you increased the number of features selected, would you get higher scores? What tradeoff would you be making if you selected more features? What other hyperparameters could you tune to improve the scores?
11. Python code that uses the Scikit-Learn Support Vector Machine algorithm LinearSVC to predict the presence of cancer based on the processed medical image data in the breast cancer dataset. Follow the same procedure as was used for Logistic Regression.
12. Outputs: State the seed value, the feature names, and the LinearSVC score for the two best and two worst combinations of features found during your testing. Don't worry about finding the optimal solution.
13. Questions: As you varied the "seed" value, what changes did you notice in the random selection of features? Were any of the features better in terms of increasing the LinearSVC score? If you increased the number of features selected, would you get higher scores? What tradeoff would you be making if you selected more features? What other hyperparameters could you tune to improve the scores?
14. Python code that uses the Scikit-Learn KNeighborsRegressor algorithm to predict diabetes progression based on a subset of medical features. Follow the same procedure as was used for the Decision Tree Regressor.
15. Outputs: State the seed value, the feature names, the number of neighbors and the KNeighborsRegressor coefficient of determination score for the two best and two worst combinations of features found during your testing. Don't worry about finding the optimal solution.
16. Questions: As you varied the "seed" value, what changes did you notice in the random selection of features? Were any of the features better in terms of increasing the k-Nearest Neighbors score? If you increased the number of features selected, would you get higher scores? What tradeoff would you be making if you selected more features? What other hyperparameters could you tune to improve the scores?
17. A brief conclusion for the project.

Main Campus students must submit their assignments on paper at the beginning of the lecture. In addition, both Internet students and Main Campus students must submit an electronic copy via SafeAssign on the ECE 563 Blackboard site BEFORE the beginning of class. SafeAssign does not allow multiple files, so you need to submit a single file, such as a PDF or a Word Document. Your file should be less than 1 MB in size. All code must be copied/pasted into your document, so we can copy/paste it into a Jupyter notebook for testing. Screenshots of code will receive zero credit.

Updated 25 Jan 2024