Collin Paiz (1576109)

Professor Anastasiu

COEN 140

13 May 2023

Program 2

Regression

Rank & F1 Score

Rank: 5

RMSE: 3.8451

Leaderboard: Visible

My Approach

My approach to this regression problem begins with initializing the problem by importing the necessary packages and reading in the *train.dat* and *test.dat* datasets. Next, I split the *train* dataset into *X_train* and *y_train*, which represents the features and the target variable, respectively, as well as set up *X_test* using the *test* dataset. After creating data frames from these matrices, I move into data cleaning. The first step I took was to fill in any missing values in the *X_train_df* and *X_test_df* data frames so that I can perform label encoding later on. After that I looped through the columns of *X_train_df* and *X_test_df* to identify which columns now had mixed value types and set those columns' value types to strings; this step was performed for label encoding, so that each column has a uniform data type. Next, I move on to validation testing to find which parameter values *k* (feature selection) and *n* (dimensionality reduction with PCA) best minimize the root mean squared error for the validation test set, created by splitting *X_train_df* and *y_train_df* into *X_tr, X_val, y_tr,* and *y_val.* After looping through different values for *k* and *n* and performing feature selection, dimensionality reduction and regression on the validation set, I can then move on with the best performing parameters for the actual predictions on the *test* dataset. In earlier iterations, I also tested out different dimensionality reduction techniques as well as different regression models (more info in the *Methodology* section). So, I end up using the *best_k* value to perform feature selection on *X_train_df* and then transform *X_test_df* accordingly. After that I use *best_n* to perform dimensionality reduction using PCA on *X_train_selected* (the data frame for *X_train_df* that resulted from feature selection) and then transform *X_test_selected* (the data frame for *X_test_df* that from feature selection). And then finally, we can fit the linear regression model using *X_train_red* (the data frame for *X_train_df* that resulted from dimensionality reduction) and *y_train_df* (the targets), which then allows us to make our predictions on *X_test_red* (the data frame for *X_test_df* that resulted from dimensionality reduction) and output the results to *output.dat*.

My Methodology

To find the best parameters and methods to use for feature selection, dimensionality reduction, and regression, I used a loop that tested different combinations of methods and parameter values and found the optimal solution based on the validation set I created by splitting the *train* dataset. I originally started with and settled for SelectKBest using f_regression because of its performance and because I often found that the alternatives yielded worse results. I also tested the use of different dimensionality reduction methods such as PCA, SVD, and FastICA. After extensive testing I found the best results came from PCA, also what I had originally started with in my earlier iterations. And lastly, I tested different regression models including Linear Regression, Ridge Regression, Lasso, Elastic Net, Decision Tree, and Random Forest. After looping through, I also found that what I had originally started with, Linear Regression, was still my best option. Because of how many loops there were, I limited my feature selection to a maximum of 20 and my dimensionality reduction to the current number of features selected. This strategy came from my first few initial tests that showed that selecting more than 20 features didn't seem to improve my results. So, this was a choice of both efficiency and accuracy. After extensive testing on the *train* dataset, I concluded that the best options are Linear Regression with PCA and SelectKBest using f_regression. After finding this out, I tweaked my train/test split size as well as my k/n value limits in a smaller loop that just looks at Linear Regression, PCA, and SelectKBest using f_regression. After making these choices, my program automatically continues on to use these methods and parameters on the test set, which yielded an RMSE of 3.8451.

Suggestions For Running

- Simply run all cells in order as is
- Results outputted to *output.dat*