Tejas Bogguram Vasudev

Cs 6375

April 29th 2023

Project 3 Report

# Pre-Processing

My input data underwent a number of pre-processing stages, including normalization, scaling, mean computation, and feature engineering.

I used the fit\_transform and transform methods of the Normalizer object from the scikit-learn package to apply it to both my training and testing data in order to normalize the input data.

Next, I used a MinMaxScaler object to scale my input data. This process scaled the input data to a predetermined range (usually between 0 and 1) and can help some machine learning algorithms perform better.

Additionally, I used the NumPy mean function to determine the mean of each row in my input data. Through this process, I was able to incorporate a new feature into the input data that shows the average value for each row.

Finally, I used the NumPy hstack function to add the mean value obtained in the previous step as a new feature to my input data.

Overall, I think these pre-processing techniques are adequate for the data I'm working with and will help me better get my data ready for machine learning algorithms.

# Polynomial model

I applied a polynomial kernel Support Vector Machine (SVM) classifier. I utilized a param\_grid dictionary with values for the regularization parameter C, the kernel type, the degree of the polynomial kernel, and the gamma parameter in order to maximize the hyperparameters of my model. I specifically took into account C values between 0.001 and 1, a polynomial kernel, a degree of 2, gamma values between 0.001 and 100, scale, auto, and six logarithmically spaced values. I used a stratified k-fold cross-validation procedure with n\_splits set to 5 to assess the performance of my model. This method was used to assess the model's performance across several data splits. I wrote out the top hyperparameters using the best\_params\_ attribute of the search object after fitting the model using the training set and hyperparameters chosen by the randomized search cross-validation. The final model was produced using the search object's best\_estimator\_ property, and it was then fitted to the training set using the fit method.

# Exponential model

A Support Vector Machine (SVM) classifier with an RBF kernel was used by me. I used a randomized search cross-validation technique with a param\_grid dictionary containing values for the regularization parameter C and the kernel type to optimize the hyperparameters of my model. I specifically took into account the RBF kernel and C values between 1 and 1,000,000. I used k-fold cross-validation to assess the effectiveness of my model. The model's performance on specific samples from the training set was evaluated , and its performance on various data splits was assessed using stratified k-fold cross-validation. I printed out the top hyperparameters using the best\_params\_ attribute of the search object after fitting the model with the training set and hyperparameters chosen by the randomized search cross-validation. The final model was produced using the search object's best\_estimator\_ attribute, and it was then fitted to the training set using the fit method.