DALHOUSIE
UNIVERSITY
Inspiring Minds

Assignment 2

Topic: SVM and KNN with performance tuning and optimization

CSCI 4155

By Shadman Mahmood

B00780608

Objective

There were two parts for this assignment. For Part A we were asked to use the “iris” dataset where we will be considering 150 samples of Iris flowers to build and train an SVM model, after performing an EDA on the dataset. Finally using GridSearchCV we were asked to tune the parameters of the model to get better results. For Part B we were asked to implement K Nearest Neighbors using the “KNN_Project_Data” and performed evaluation methods to check results. Finally, we performed Elbow method to improve the performance of our model by picking an optimal K value.

Detailed Process

Part A

We were asked to display pictures of the 3 types of iris flowers with dimensions 300*300 we were given urls for the 3 species in teams, so I used a urllib to retrieve the images and display them using matplotlib by setting figsize to (300,300) I was able to specify with the given dimensions. Next we were asked to load the dataset using seaborn, sns.load. After performing the EDA on the dataset, it was clear by the pairplots that iris setosa was the easiest one to separate out of the other species, as it had the highest petal and sepal length, with a largest petal width and the shortest sepal width. The other two species values were fairly close to each other. Looking at the KDE plot it shows the probability density distribution where the petal length ranges from 0.9 to 2 but most of the values of petal length and width for setosa are concentrated 1.2 to 1.7 for the petal length and from 1 to 0.4 for petal width. After building and training the SVM model the model evaluation showed an accuracy of 1 so after performing GridSearchCV I still could not improve the result.

Part B

We were asked to import the KNN_Project_Data as the dataset and perform some analysis for which I chose the methods:

- display(knn_df.head(5)) = display the first 5 rows of the dataset
- knn_df.info() and .describe() to get the mean, sum, dimension of data
- knn_df.isnull().sum() = I used this to check if there were any missing values in the dataset

for the EDA I created a pairplot and heatmap which showed that the data was correlated with each other. I performed standardization over all the data except the TARGET CLASS and verified it by printing it out. After building and training the KNN model I created a confusion matrix and classification report which showed an accuracy of 74 percent. To perform elbow method, we first had to create a list of error rates using a loop and training the model by variable n neighbors between range 1 to 40 as depicted in the graph given in our assignment. The predicted values were checked with Y_test and recorded in the error rate list. Next, I created a plot with an identical style to the sample plot supplied to us. By looking at the plot I generated I picked K values as 21 because after this point there are no major changes in performance and higher K values will only result in taking up more resources. By taking K as 21 we can see an accuracy boost of 12 percent, previously the accuracy was 0.72 and new accuracy is 0.84.