Task 3

ACM Summer School 2021, Artificial Intelligence

Mentors: Mobina Shahbandeh, Amirhossein Abaskohi,
Mahdi Mehmanchi, Erfan Mirzaei

1 Introduction

In this task we are going to examine SVM, Decision Tree, and Random forest algorithms on a dataset.

2 Support Vector Machine

A Support Vector Machine (SVM) is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection.

2.1 Reading the dataset and preprocessing

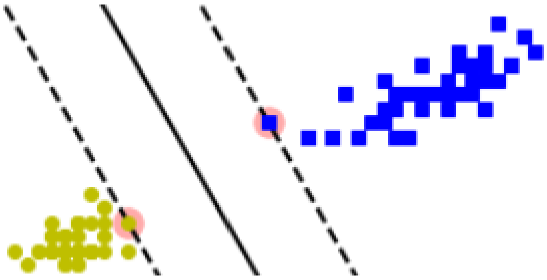
Here we are going to use *iris dataset*. The dataset is provided to you. Use *Numpy* or *Pandas* to read the dataset and then process the data. Depending on the outcome, you may need to repeat this step to achieve the optimum result.

2.2 Data visualization

Now using *Matplotlib*, show the **Versicolor** and **Setosa** groups in a 2D plot. The features you should use to plot the groups are **petal length** and **petal width**.

2.3 Train a binary SVC

Use Support Vector Classifier(SVC) from *sklearn* library with **linear** kernel to train a classifier. After training your classifier, now visualize your data with the line which is predicted for your dataset. You should plot the margin lines as well. This is how your plot should appear(the colors and the shapes do not matter):



2.4 Checking sensitivity to scaling

SVMs are sensitive to feature scales. To check this, use *StandardScaler* from *sklearn* library and use **fit transform** on your input and then train your SVC again.

2.5 Visualizing the whole data

Now use *pairplot* from *seaborn* library to plot the whole dataset. Do you think they are linearly separable?

2.6 Train/Test split

Now use *train test split* function from *sklearn* to split your dataset to train and test sets.

2.7 Train your SVC

Now create an SVC. Let the hyper-parameters be as their default values. Train your model on your train set. After training, you can see your SVC hyper-parameters. In your report discuss all of them.

2.8 Model evaluation

Now predict the test set. Search for confusion matrix, precision, recall, and f1 score. Now use *confusion matrix* and *classification report* to check your model performance.

2.9 Tune your hyper-parameters

Now use *GridSearchCV* to tune your hyper-parameters.

3 Decision Tree and Random Forest

Like SVMs, Decision Trees are versatile Machine Learning algorithms that can perform both classification and regression tasks, and even multioutput tasks. They are very powerful algorithms, capable of fitting complex datasets.

Decision Trees are also the fundamental components of Random Forests, which are among the most powerful Machine Learning algorithms available today.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

3.1 Train a decision tree

Train a Decision Tree Classifier from *sklearn* library. Let the hyper-parameters be their default values.

3.2 Plot your decision tree

Now use **plot tree** from *matplotlib* library to plot your tree.

Train your model again setting the **max depth** to **three** and then plot your tree again.

What is the difference?

3.3 Tune your hyper-parameters

Now use *GridSearchCV* to tune your hyper-parameters.

3.4 Evaluate your model

Use accuracy, recall, precision, and f1 score to evaluate your model on your test set.

3.5 Train a random forest

Build and train a random forest classifier from *sklearn* library on your train dataset.

3.6 Evaluate your model

Use accuracy, recall, precision, and f1 score to evaluate your model on your test set.

3.7 Comparing random forest and decision tree

In your report, compare the best models you have made.

4 Try on previous assignment's dataset(Bonus)

Test these algorithms on the admission dataset you had in your previous assignment now that you know how to construct Decision Tree, Random Forest, and SVM. Compare the outcomes.