COS40007 Artificial Intelligence for Engineering

Week 3 Studio Activities

ILO	Understand ML model development and Hyper parameter tuning.
Aim	• Learn how perform merging and splitting operations on data • Learn how to train different ML models • Learn how to improve performance of ML model through hyper parameter tuning. • Learn how to perform feature selection and dimensionality reduction
Resources	Books: 1. Prosise, Jeff. Applied machine learning and AI for engineers. " O'Reilly Media, Inc.", 2022. 2. Raschka, Sebastian, Yuxi Hayden Liu, and Vahid Mirjalili. Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python. Packt Publishing Ltd, 2022. Web Resources: 1. https://www.kaggle.com/code/faressayah/support-vector-machine-pca-tutorial-for-beginner 2. https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/
Requirements for to be marked as complete	Demonstrate the table of outcome to your tutor

Disclaimer: The dataset used in this studio was originally collected for a funded research project by Australian Meat processor Corporation. The dataset here is used solely for educational purposes and can be only used for completing activities of this studio. By any mean this dataset is not shareable to others or any public domain.

Dataset: The dataset used in this studio was collected from an Australian meat processing plant in real world settings. The data was collected using hand motion sensor of meant plant workers. The raw data from sensors contains acceleration and gyroscope in three-dimensional space. The dataset you got in this studio is a pre-processed version of this raw data that contains 156 features extracted from motion data of sensors from 2 hands (left and right). The purpose of the data is to understand

different activities performed by the meant plant worker during their working shift for cutting meat. The dataset contains 4 CSV files which is collected from activities of 4 meant plant workers during their working shifts (so, w1 refer to worker 1, w2 refere to worker2 and so on). The last column of the dataset is a class of activity. Here 3 activities are defined, 0 -> idle, 1 -> worker is in work (e.g., cutting meat), 2-> worker is sharpening cutting knife. So, this is your target variable or class variable.

Hyperparameters

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. However, there are some parameters, known as Hyperparameters and those cannot be directly learned. They are commonly chosen by humans based on some intuition or hit and trial before the actual training begins. These parameters exhibit their importance by improving the performance of the model such as its complexity or its learning rate. Models can have many hyper-parameters and finding the best combination of parameters can be treated as a search problem.

Dimensionality Reduction

Dimensionality reduction is a technique used to reduce the number of features in a dataset while retaining as much of the important information as possible. In other words, it is a process of transforming high-dimensional data into a lower-dimensional space that still preserves the essence of the original data.

In machine learning, high-dimensional data refers to data with a large number of features or variables. The curse of dimensionality is a common problem in machine learning, where the performance of the model deteriorates as the number of features increases. This is because the complexity of the model increases with the number of features, and it becomes more difficult to find a good solution. In addition, high-dimensional data can also lead to overfitting, where the model fits the training data too closely and does not generalize well to new data.

Dimensionality reduction can help to mitigate these problems by reducing the complexity of the model and improving its generalization performance. There are two main approaches to dimensionality reduction: feature selection and feature extraction.

Studio Activity 1: Data preparation

1) Download the provided dataset (ampc.zip) from canvas and extract this. You will find 4 csv files after unzipping the file. This is a pre-processed data that contains 157 columns. First 156 columns are computed features from raw data and the last one is the class label.
2) Combine the 4 CSV files to a single CSV file (combined_data.csv) [you can use pandas merge/contact/append or similar operation do this]
3) Next you shuffle the data and save in another CSV file (all_data.csv)

Studio Activity 2: Model Training

Using "all_data.csv

1) Sperate feature and class as X and y
2) Train a svm model using
    a. Splitting train and test set to 70% and 30% and measure the model accuracy

Sample code

```
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=1)
clf = svm.SVC()
clf.fit(_X_train, y_train)
y_pred = clf.predict(X_test)
accuracy_score(y_test,y_predict)
```

    b. 10-fold cross validation and measure model accuracy (cross validation score)

Sample code

```
from sklearn.model_selection import cross_val_score
from sklearn import svm

clf = svm.SVC()
scores = cross_val_score(clf, X, y, cv=10)
print (scores)
```

save the classification accuracy of the above 2 cases

Studio Activity 3: Hyper parameter tuning

Now use this reference link to perform hyperparamenter tuning on your dataset

1) By default SVC use liner kernel, use rbf kernel instead
2) Use GridSearchCV to identify optimal values of hyper parameters
3) Now use the optimal values identified in GridSearchCV to update your SVM model in Activity 2 and obtain classification accuracy for both train-test split and 10-fold cross validation

Studio Activity 4: Feature Selection

Use 100 best (using k best feature selection method described here) features and generate result of another 2 SVM model with

a) 70/30 train/test set split with hyperparameter tuning (using values obtained in activity 3)
b) 10-fold class validation with hyperparameter tuning (using values obtained in activity 3)

Studio Activity 5: Dimensionality reduction

Use Principal Component Analysis (PCA) for to reduce dimension on your data. Take first 10 principal components as features and again train 2 SVM models

a) 70/30 train/test set split with hyperparameter tuning (using values obtained in activity 3)
b) 10-fold class validation with hyperparameter tuning (using values obtained in activity 3)

Sample steps:

1. pca = PCA().fit(X) [available in from sklearn.decomposition import PCA]
2. Now to take 10 principle components featured use pca.components list
3. Then use that 10 principal components feature as new X for training SVM model

Studio Activity 6: Prepare a summary table

Now prepare a summary table containing accuracy value of SVM models you developed

SVM model	Train-test split	Cross validation
Original features	XX%	XX%
With hyper parameter tuning	XX%	XX%
With feature selection and hype parameter tuning	XX%	XX%
With PCA and hyper parameter tuning	XX%	XX%

Studio Activity 7: Other classifiers

Use the original data (all_data.csv) to

1. Train with SGDclassifier for both train-test split and cross-validation and obtain the accuracy value
2. train with RandomForest for both train-test split and cross-validation and obtain the accuracy value
3. train with MLPclassifier for both train-test split and cross-validation and obtain the accuracy value

Finally prepare another summary table and accuracy like the following

Model	Train-test split	Cross validation
SVM	XX%	XX%
SGD	XX%	XX%
RandomForest	XX%	XX%
MLP	XX%	XX%

Next Steps:
The assessment Task for Week 3 now can be attempted and submitted via Canvas.