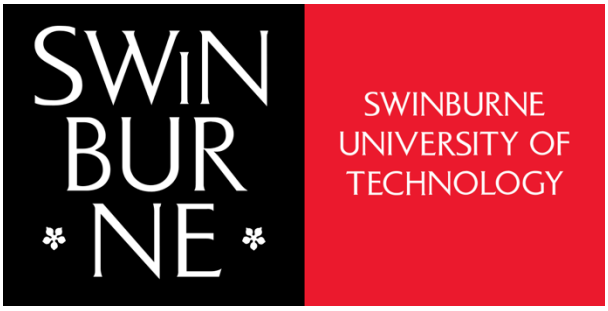

Artificial Intelligence for Engineering

Portfolio Assessment 2

Systematic Approach to Develop ML Model

Xuan Tuan Minh Nguyen - 103819212
Studio 1-6

Summary Table of Studio 3

1. Summary Table of Activity 6

SVM Model	Train-test split	Cross-validation**
Original Features	88.22%	89.17%
With hyper-parameter tuning*	83.66%	84.35%
With feature selection and hyper-parameter tuning*	84.87%	85.61%
With PCA and hyper-parameter tuning*	83.75%	84.41%

Table 1: Summarization of accuracy values and cross-validation values of developed SVM models

**: The tuned hyper-parameter setting is obtained by using GridSearchCV, which obtained a C = 10, gamma = 0.0001, kernel = 'rbf'*
***: The 10-fold cross-validation values*

2. Summary Table of Activity 7

Model	Train-test split	Cross-validation**
SVM	88.22%	89.17%
SGD	88.95%	87.14%
RandomForest	92.22%	92.67%
MLP	86.72%	86.22%

Table 2: Summarization of accuracy values and cross-validation values of different models

***: The 10-fold cross-validation values*

3. Source Code

- Github: https://github.com/cobeo2004/COS40007/blob/main/Studio%203

Source code and data for each step

1. Full code

- Step 1 + 4 Python Notebook: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/Assignment2.ipynb
- Step 1 + 4 Python: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/assignment2-all.py
- Step 1 + 3: Data: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/ampc2/processed_all_data.csv

2. Step 1: Data collection

- Source Code: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/assignment2-s1.py
- Data: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/ampc2/combined_data.csv

3. Step 2: Create composite columns

- Source Code: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/assignment2-s2.py
- Data: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/ampc2/composited_data.csv

4. Step 3: Data pre-processing

- Source Code: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/assignment2-s3.py
- Data: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/ampc2/processed_all_data.csv

5. Step 4: Training

- Source Code: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/Assignment2.ipynb

- Data: https://github.com/cobeo2004/COS40007/blob/main/Assignment%202/ampc2/processed_all_data.csv

Training outcome and Model Selection

1. *Training (outcome summary table)*

Models	Train-test split	Cross-validation**
SVM with Original Features	97.78%	96.01%
SVM with hyper-parameter tuning*	77.29%	75.19%
SVM with feature selection and hyper-parameter tuning*	77.29%	75.19%
SVM with PCA and hyper-parameter tuning*	79.25%	75.19%
SGD	93.63%	91.10%
RandomForest	99.72%	96.59%
MLP	94.18%	94.01%

Table 3: Summarization of accuracy values and cross-validation values of models

**: The tuned hyper-parameter setting is obtained by using GridSearchCV, which obtained a C = 0.1, gamma = 1, kernel = 'rbf'*
***: The 10-fold cross-validation values*

2. *Model Selection*

Here are some of my model selection decisions based on the given accuracy score:

- The best SVM model for my problem is the "**SVM with Original Features**" is the best SVM model, which achieves 97.78% in split accuracy and 96.01% in cross-validation accuracy. It yields a higher accuracy when compared with other SVM variants, which uses hyper-parameter tuning, feature selection with SelectKBest and PCA to reduce dimensionality. Thus

making it effectively in capturing the key patterns in the boning and slicing datasets.

- The best Machine Learning model for my problem is the **Random Forest Model,** which has a 99.72% in split accuracy and 96.59% in cross-validation accuracy. It surpasses all SVM models and the rest of the Machine Learning models like SGD (93.63% and 91.10%) and MLP (94.18% and 94.01%). Thus, this model ensures an optimal choice for classifying tasks related to the performance metrics.