UTM UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECB3203-01(PROGRAMMING FOR BIOINFORMATIC)

Semester 01 2025/2026
Section 01

Progress 4

Faculty of Computing

Dataset:
https://www.kaggle.com/datasets/uciml/mushroom-classification

---

STUDENT NAME	MATRIC NUMBER
NURUL NATALIA BINTI ROSNIZAM	A23CS0165
ADLINA NARISYA BINTI ISMAIL	A23CS0033
NAZATUL NADHIRAH BINTI SABTU	A23CS0144

4.0 Model Development

The primary objective of Model Development is to develop, train, and prepare a machine learning classification models that capable of accurately predicting mushroom edibility based on the selected features. The goal is to implement and train several models (Random Forest, Decision Tree, and Logistic Regression) to establish a baseline for performance evaluation in the subsequent phase.

4.1 Data Partitioning (Train-Test Split)

To avoid biased model evaluation, the prepared dataset was divided into two subsets, a training set and a testing set. This was accomplished by the train_test_split() function from the scikit-learn library. The model trains on a portion of the data and objectively evaluating its performance on unseen data. This helps prevent overfitting where the model only memorizes the training data but performs poorly using the new or unseen data and underfitting.

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.25, random_state=42)
```

Training Set (75%): This subset consists of xtrain and ytrain, used to train the models. The models analyze this data to learn the algorithms and relationships between the features and the target class.

Testing Set (25%): This subset consists of xtest and ytest, and was kept separate during the training process. It serves as unseen data to evaluate the generalization performance of the trained models fairly.

A random_state = 42 was used to ensure that the data split was identical every time the code was run.

4.2 Model Training

A diverse set of three classification algorithms was chosen:

4.2.1 Random Forest Classifier

```python
cla = RandomForestClassifier(n_estimators = 10, random_state = 42)
cla.fit(xtrain, ytrain)
```

A Random Forest Classifier model is instantiated. A method operates by constructing a number of decision trees during training. n_estimators = 10 means the forest will be composed of 10 decision trees. Known for its high accuracy, and robustness to overfitting.

4.2.2 Decision Tree Classifier

```python
dst = DecisionTreeClassifier()
dst.fit(xtrain, ytrain)
```

A simple, non-parametric machine learning algorithm that serves as an excellent baseline. It uses a tree-like structure to make a decision and classify data into categories. Split the dataset into smaller, more homogeneous subsets based on the most significant features until leaf nodes (terminal nodes) are decided.

4.2.3 Logistic Regression

```python
lr = LogisticRegression()
lr.fit(xtrain, ytrain)
```

A linear model used for binary classification. This model examines the relationship between all variables and target variables. It is a linear model, thus it used when the output variable or target variable is continuous.

All selected models were instantiated and trained using the .fit() function. Each model trained using the training data (xtrain, ytrain), allowing it to learn and know the pattern of mapping from input features to the target output. The models also independent during the training process by trained separately using the same training data.