HW3 - Predictive Analytics

Sathvika Bhagyalakshmi Mahesh
Sb8913

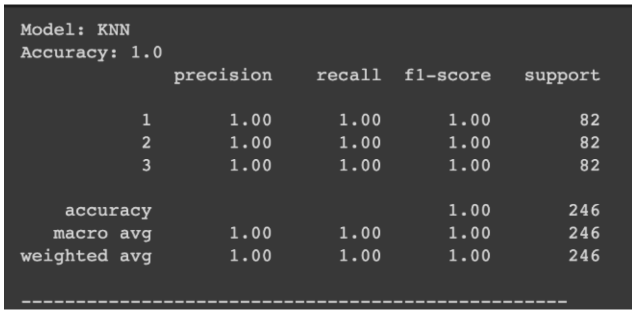
STEP 1: Problem Statement:
Lung cancer prediction - Q1 - **Lung_Cancer Prediction** - Lung Cancer Prediction Air Pollution, Alcohol, Smoking & Risk of Lung Cancer
The **dataset** contains information on patients with lung cancer, including their age, gender, air pollution exposure, alcohol use, dust allergy, occupational hazards, genetic risk, chronic lung disease, balanced diet, obesity, smoking, passive smoker, chest pain, coughing of blood, fatigue, weight loss ,shortness of breath ,wheezing ,swallowing difficulty ,clubbing of finger nails and snoring

STEP 2: Data Sources:
`Here is the website -`
`https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link/data`

Lung Cancer Prediction - Air Pollution, Alcohol, Smoking & Risk of Lung Cancer

STEP 3: Feature engineer and preprocessing
Step 3: Data cleaning, Validation, Preprocessing
Step a: Handling Missing Values - Dropping rows with any missing values
Step b: Removing Duplicates - Dropping duplicate rows if any
Step c: Correcting Data Errors - This step is subjective and might require domain knowledge
Step d: Data Cleaning - Identifying Outliers
Step 3 : Data Preprocessing
Before Train Test Split, split X(Input) & y(target) from the original Dataframe
Standardization
Applying Principal Component Analysis (PCA) for dimensionality Reduction
Modeling - KNN, SVM, Decision Tree, Random Forest, Gradient Boosting

STEP 4: Modelling

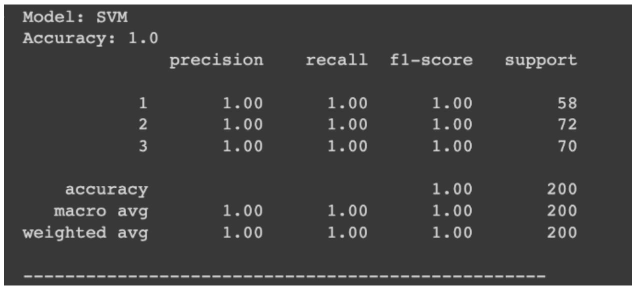


```
Model: KNN
Accuracy: 1.0
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        82
           2       1.00      1.00      1.00        82
           3       1.00      1.00      1.00        82

    accuracy                           1.00       246
   macro avg       1.00      1.00      1.00       246
weighted avg       1.00      1.00      1.00       246

--------------------------------------------------
```

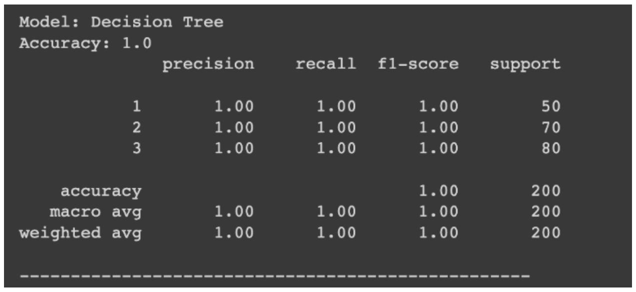


```
Model: SVM
Accuracy: 1.0
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        58
           2       1.00      1.00      1.00        72
           3       1.00      1.00      1.00        70

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200

--------------------------------------------------
```

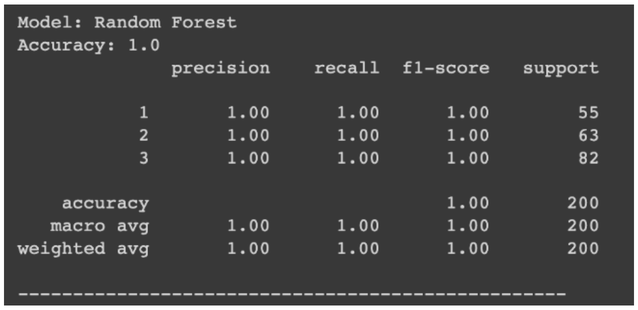


```
Model: Decision Tree
Accuracy: 1.0
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        50
           2       1.00      1.00      1.00        70
           3       1.00      1.00      1.00        80

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200

--------------------------------------------------
```

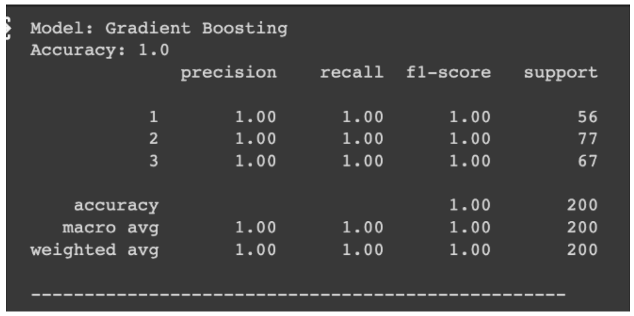


```
Model: Random Forest
Accuracy: 1.0
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        55
           2       1.00      1.00      1.00        63
           3       1.00      1.00      1.00        82

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200

--------------------------------------------------
```



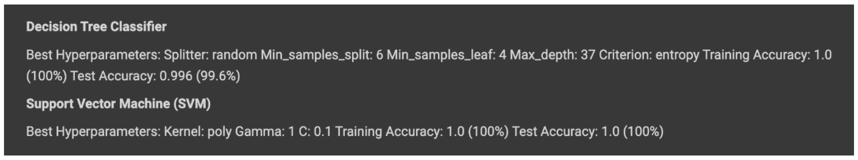
```
Model: Gradient Boosting
Accuracy: 1.0
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        56
           2       1.00      1.00      1.00        77
           3       1.00      1.00      1.00        67

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200

--------------------------------------------------
```

Step 5: Hyper-parameter
Hyperparameter Tuning - SVC and Decision Tree
Random Search is performed and Hyper Parameter Tuning - SVM and Decision Tree



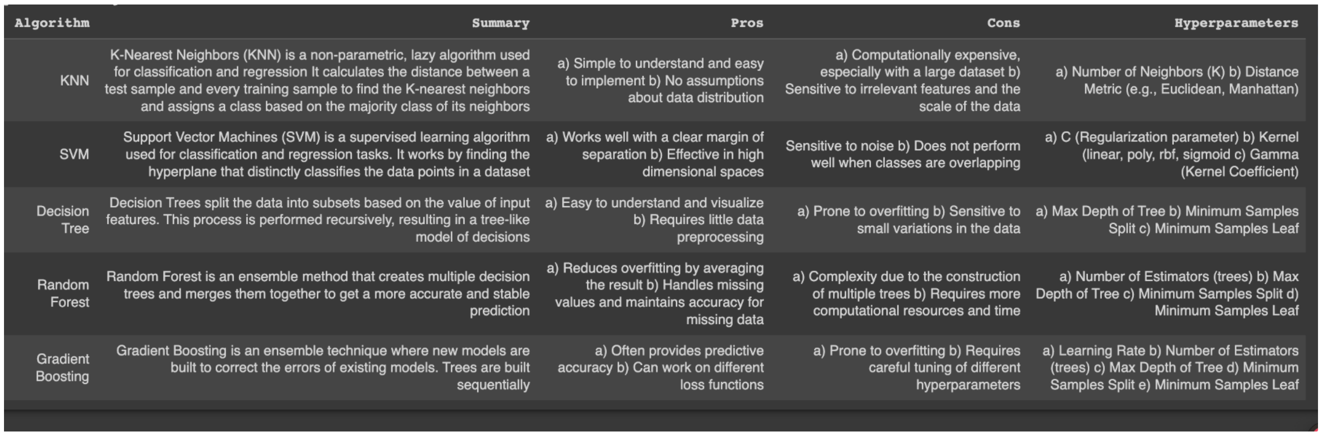
Decision Tree Classifier

Best Hyperparameters: Splitter: random Min_samples_split: 6 Min_samples_leaf: 4 Max_depth: 37 Criterion: entropy Training Accuracy: 1.0 (100%) Test Accuracy: 0.996 (99.6%)

Support Vector Machine (SVM)

Best Hyperparameters: Kernel: poly Gamma: 1 C: 0.1 Training Accuracy: 1.0 (100%) Test Accuracy: 1.0 (100%)

Step 6: Conclusion



Algorithm	Summary	Pros	Cons	Hyperparameters
KNN	K-Nearest Neighbors (KNN) is a non-parametric, lazy algorithm used for classification and regression It calculates the distance between a test sample and every training sample to find the K-nearest neighbors and assigns a class based on the majority class of its neighbors	a) Simple to understand and easy to implement b) No assumptions about data distribution	a) Computationally expensive, especially with a large dataset b) Sensitive to irrelevant features and the scale of the data	a) Number of Neighbors (K) b) Distance Metric (e.g., Euclidean, Manhattan)
SVM	Support Vector Machines (SVM) is a supervised learning algorithm used for classification and regression tasks. It works by finding the hyperplane that distinctly classifies the data points in a dataset	a) Works well with a clear margin of separation b) Effective in high dimensional spaces	Sensitive to noise b) Does not perform well when classes are overlapping	a) C (Regularization parameter) b) Kernel (linear, poly, rbf, sigmoid c) Gamma (Kernel Coefficient)
Decision Tree	Decision Trees split the data into subsets based on the value of input features. This process is performed recursively, resulting in a tree-like model of decisions	a) Easy to understand and visualize b) Requires little data preprocessing	a) Prone to overfitting b) Sensitive to small variations in the data	a) Max Depth of Tree b) Minimum Samples Split c) Minimum Samples Leaf
Random Forest	Random Forest is an ensemble method that creates multiple decision trees and merges them together to get a more accurate and stable prediction	a) Reduces overfitting by averaging the result b) Handles missing values and maintains accuracy for missing data	a) Complexity due to the construction of multiple trees b) Requires more computational resources and time	a) Number of Estimators (trees) b) Max Depth of Tree c) Minimum Samples Split d) Minimum Samples Leaf
Gradient Boosting	Gradient Boosting is an ensemble technique where new models are built to correct the errors of existing models. Trees are built sequentially	a) Often provides predictive accuracy b) Can work on different loss functions	a) Prone to overfitting b) Requires careful tuning of different hyperparameters	a) Learning Rate b) Number of Estimators (trees) c) Max Depth of Tree d) Minimum Samples Split e) Minimum Samples Leaf

Github link: https://github.com/sathvikabm/HW3-PredictiveAnalytics

Steps performed

1) a)data cleaning b) validating c) preprocessing
2) Benchmark various algorithms
    a) KNN implementation from scratch no inbuilt libraries
    b) SVM
    c) Decision Trees
    d) Random Forest
    e) Gradient Boosting
3) Perform hyperparameter searches - 2 algorithm - SVM and Decision Tree