Project 1
Machine Learning
Diabetes Prediction
Report

Net ID: CRR220000 Name: Chetan Rajendra Rahane

Introduction

The objective of this project was to build a machine learning model to predict the likelihood of diabetes in individuals based on health and lifestyle attributes. The dataset used consisted of 50,000 rows and a wide range of features such as blood pressure, cholesterol levels, physical activity, and general health. To achieve optimal predictive performance, I experimented with multiple machine learning algorithms, including **Decision Tree** and **Random Forest** classifiers, and employed hyperparameter tuning to enhance the models' accuracy.

After thorough experimentation with different settings and cross-validation techniques, along with GridSearchCV, I concluded that the **Random Forest Classifier** provided superior performance compared to the Decision Tree model. This report outlines the methodology used, the tuning process, and the reasoning behind selecting the final model.

Model Training and Tuning Process

1. Dataset Preprocessing

The dataset was preprocessed to handle missing values and encode categorical variables. No feature scaling was applied as decision tree-based algorithms, including Random Forest, do not require scaling due to their nature of splitting data on thresholds of individual features.

2. Model Selection

I began by experimenting with two machine learning algorithms:

- **Decision Tree Classifier**: This was chosen for its simplicity and interpretability.
- **Random Forest Classifier**: An ensemble learning method that constructs multiple decision trees and outputs the mode of their predictions, offering improved accuracy and robustness against overfitting.

3. Hyperparameter Tuning

For each classifier, I performed hyperparameter tuning using **GridSearchCV** to find the best parameters based on K-fold cross-validation(K=10,20). The parameter grid (param_grid) for both classifiers included options to tune attributes like maximum tree depth, minimum samples per split, and the splitting criterion(example written below).

For Decision Tree:

- param_grid = {'criterion': ['entropy'], 'max_depth': [2, 4, 5, 6, 8, 10, 20, 30, None], 'min_samples_split': [2, 10, 20], 'min_samples_leaf': [1, 4, 8]}

After testing different parameter combinations, the best parameters for the Decision Tree were:

- {'criterion': 'entropy', 'max_depth': 7, 'min_samples_leaf': 3, 'min_samples_split': 2}
- Accuracy: 74.06%

For Random Forest:

- param_grid = {'n_estimators': [50, 100, 200, 300], 'criterion': ['entropy'], 'max_depth': [3, 5, 10, 20, 30, None], 'min_samples_split': [2, 5, 10, 20], 'min_samples_leaf': [1, 2, 4, 8]}

After testing different parameter combinations, the best-performing Random Forest parameters were:

- {'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 8, 'min_samples_split': 20, 'n_estimators': 100}
- Accuracy: 75.16%

Explanation of Results

1. Decision Tree Classifier

The Decision Tree model provided accuracy of around 74.06%, especially when the depth of the tree was not controlled. The grid search process revealed that using criterion='entropy' and limiting the tree's depth to 7 with min_samples_leaf=3 and min_samples_split=2 helped in reducing overfitting while maintaining reasonable accuracy. The Confusion matrix looked similar to:

848	391
255	1006

2. Random Forest Classifier

Random Forest classifier performed a bit better. By averaging across a set of decision trees, the Random Forest algorithm reduced overfitting and thus generalized a bit better on unseen data. The parameters chosen, max_depth=20, min_samples_leaf=8, and min_samples_split=20, prevented the growth of individual trees from being too deep, since too deep could lead to overfitting, and not so shallow as to underfit. The confusion matrix looked similar to:

879	340
280	1001

Conclusion

After conducting multiple rounds of hyperparameter tuning and cross-validation, the **Random Forest Classifier** was found to be the superior model for this dataset. It provided better accuracy and generalization than the Decision Tree, due to its ensemble nature and robustness in capturing complex patterns without overfitting.

Key takeaways:

- Random Forest achieved the highest accuracy with the parameters: {'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 8, 'min_samples_split': 20, 'n_estimators': 100}.
- Decision Tree, although simpler and easier to interpret, had lower performance than Random Forest due to its susceptibility to overfitting.

Thus, the **Random Forest Classifier** with the selected parameters was chosen as the final model for the task of predicting diabetes likelihood. This model is both accurate and efficient, making it a solid choice for deployment in real-world predictive applications.