Ikhlaq Ahmad

Dr. Crystal Muang

CS 6375.005

October 9, 2024

Project 1: Random Forest and Decision Tree Classification

**Overview**

The goal of this project is to classify individuals as either diabetic or non-diabetic based on a dataset with 21 features. Two models were employed: DecisionTreeClassifier and RandomForestClassifier from the scikit-learn library.

**Libraries:**

* Scikit-learn
* Pandas
* Pickle
* Seaborn
* Matplotlib

Specific versions and complete list in the requiremets.txt file

**Data Preprocessing:**

* Data Loading: The dataset was loaded using pandas.
* Feature Selection: Categorical and numerical features were identified. Categorical features were analyzed separately for their relationship with the target variable, while numerical features were standardized.
* Data visualization: Data was graphically visualized using Matplotlib for every feature and their related distribution(s). More details in the proj1\_evaluate.ipynb file.
* Standardization: The StandardScaler was used to standardize numerical features for both training and testing datasets to ensure that all features contributed equally to the model training.

**Model Training:**

* Model Selection: DecisionTreeClassifier and RandomForestClassifier were chosen for their effectiveness in handling classification problems.
* Grid Search: A grid search was conducted to optimize model hyperparameters, including: max\_depth, max\_samples\_split, etc.
* criterion: The function to measure the quality of a split (e.g., 'gini', 'entropy').
* n\_estimators (for Random Forest): The number of trees in the forest.
* Cross-Validation: A 5-fold cross-validation was utilized to assess the model's performance and to avoid overfitting. This process provided mean and standard deviation for the accuracy scores, giving a robust evaluation of the model's performance.

Grid Search and Cross-Validation Results

The best parameters obtained through grid search and their corresponding cross-**Validation scores:**

**Decision Tree: Accuracy: 0.7357**

* Best Decision Tree parameters: {'criterion': 'entropy', 'max\_depth':10, 'min\_samples\_leaf': 5, 'min\_samples\_split': 2}
* Best Decision Tree Cross-Validation Accuracy: 0.7329
* Decision Tree Mean Accuracy: 0.7329
* Decision Tree Accuracy Standard Deviation: 0.0057

**Random Forest: Accuracy: 0.7524**

* Best Random Forest parameters: {'bootstrap': True, 'class\_weight': 'balanced', 'criterion': 'gini', 'max\_depth': 10, 'max\_features': 'sqrt', 'n\_estimators': 100, 'n\_jobs': -1, 'random\_state': 42}
* Best Random Forest Cross-Validation Accuracy: 0.7537
* Random Forest Mean Accuracy: 0.7537
* Random Forest Accuracy Standard Deviation: 0.0075

**Post-Training Data Visualization:**

Confusion Matrix was utilized to display reults of both models.

**Testing File:**

The testing file has been modified. In the file, it contains a section “Changed Code”. The code basically preprocesses the testing data for standardization.

############## Changed Code ##################################

# Preprocessing: Standardizing the features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X) # Scale the test data

model = load\_model(model\_filename)

total\_right = 0

total\_wrong = 0

Y\_pred = model.predict(X\_scaled)

model = load\_model(model\_filename)

############## Changed Code ##################################

**Run the file:**

* pip install -r requirements.txt

**Model:**

* best\_rf\_model.pkl

**Python and scikit versions:**

* Python 3.12.6
* Sklearn 1.5.2

**Folder:**

* project1.pdf
* project1.ipynb
* requirement.txt
* proj1\_evaluate.py
* Evaluation.docx
* best\_rf\_model.pkl