VISHWAKARMA UNIVERSITY
Maximising Human Potential

Activity based

Project 1 Report on Datawarehouse and Data Minning

Submitted to Vishwakarma University, Pune

Under the Initiative of

Contemporary Curriculum, Pedagogy, and Practice (C2P2)

By

Atharva Shevate

SRN No : 202201727

Div : E

Second Year Engineering

Department of Computer Engineering

Faculty of Science and Technology

Academic Year

2023-2024

Project title :

Use Decision Tree classification on a dataset in the healthcare domain to predict disease occurrence based on patient medical records.

Project Objective:

The primary objective of this project is to build a machine learning model using the Decision Tree classification algorithm to predict the occurrence of specific diseases based on patient medical records. The model aims to assist healthcare providers in identifying potential high-risk patients, enabling early diagnosis and personalized treatment plans**.**

Project Description:

This project focuses on applying the Decision Tree algorithm to a healthcare dataset that contains patient medical records, including attributes such as age, gender, blood pressure, cholesterol levels, and other relevant medical factors. The steps involved in the project include:

1. **Data Collection**: A healthcare dataset containing anonymized patient records is collected. Each record includes various medical parameters, along with a label indicating whether or not a specific disease occurred.
2. **Data Preprocessing**: The dataset is cleaned and preprocessed to handle missing values, normalize or scale continuous data, and encode categorical variables.
3. **Feature Selection**: Important features that contribute to disease prediction are selected based on medical relevance and statistical analysis.
4. **Model Training**: A Decision Tree classifier is trained using the dataset. The model splits the data based on key attributes to classify whether a patient is likely to have a certain disease.

5. **Model Evaluation**: The model is evaluated using accuracy, precision, recall, and F1 score metrics. Cross-validation techniques are employed to ensure the model's robustness.
6. **Prediction**: The trained model is used to predict the occurrence of disease in new, unseen patient records.

SOURCE CODE:

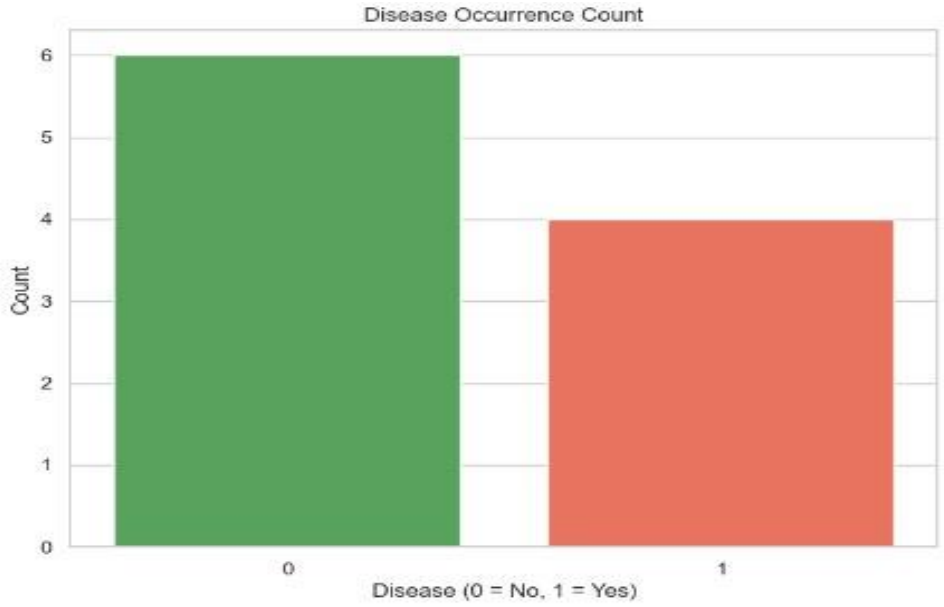
```
# Importing Necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.tree import export_text

# Setting Seaborn Style
sns.set(style="whitegrid")

# Sample Dataset (For demonstration, replace with your actual healthcare dataset)
# Data contains features like age, blood pressure, cholesterol, and labels for disease presence
data = {
   'Age': [25, 30, 45, 35, 50, 23, 43, 55, 37, 29],
   'Blood_Pressure': [120, 130, 140, 135, 145, 118, 138, 150, 136, 125],
   'Cholesterol': [180, 200, 250, 220, 230, 175, 245, 260, 215, 190],
   'Diabetes': [0, 0, 1, 0, 1, 0, 1, 1, 0, 0],  # 0: No Diabetes, 1: Diabetes
   'Disease': ['No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No', 'No']
}

# Load data into a DataFrame
df = pd.DataFrame(data)

# Displaying the initial dataset
```

```
print(df)

# Converting Categorical Column (Disease) into Numerical Column
label_encoder = LabelEncoder()
df['Disease'] = label_encoder.fit_transform(df['Disease'])

# Data Visualization
plt.figure(figsize=(8, 6))
sns.countplot(x='Disease', data=df, palette=['#4CAF50', '#FF6347'])
plt.title("Disease Occurrence Count")
plt.xlabel("Disease (0 = No, 1 = Yes)")
plt.ylabel("Count")
plt.show()

# Splitting the dataset into features (X) and target (y)
X = df[['Age', 'Blood_Pressure', 'Cholesterol', 'Diabetes']]
y = df['Disease']

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Feature Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Building Decision Tree Classifier Model
dt_classifier = DecisionTreeClassifier(criterion='entropy', max_depth=4, random_state=42)
dt_classifier.fit(X_train_scaled, y_train)

# Making Predictions
y_pred = dt_classifier.predict(X_test_scaled)

# Evaluating the Model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy * 100:.2f}%")
print("\nConfusion Matrix:")
print(conf_matrix)
```

```
print("\nClassification Report:")
print(classification_rep)

# Visualization of Confusion Matrix
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
        xticklabels=['No Disease', 'Disease'], yticklabels=['No Disease', 'Disease'])
plt.title("Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()

# Plotting the Decision Tree
plt.figure(figsize=(14, 8))
plot_tree(dt_classifier, feature_names=['Age', 'Blood_Pressure', 'Cholesterol', 'Diabetes'],
      class_names=['No Disease', 'Disease'], filled=True, rounded=True, fontsize=10)
plt.title("Decision Tree Visualization")
plt.show()

# Text Representation of Decision Tree Rules
tree_rules = export_text(dt_classifier, feature_names=['Age', 'Blood_Pressure', 'Cholesterol',
'Diabetes'])
print("\nDecision Tree Rules:\n")
print(tree_rules)
```

OUTPUT:

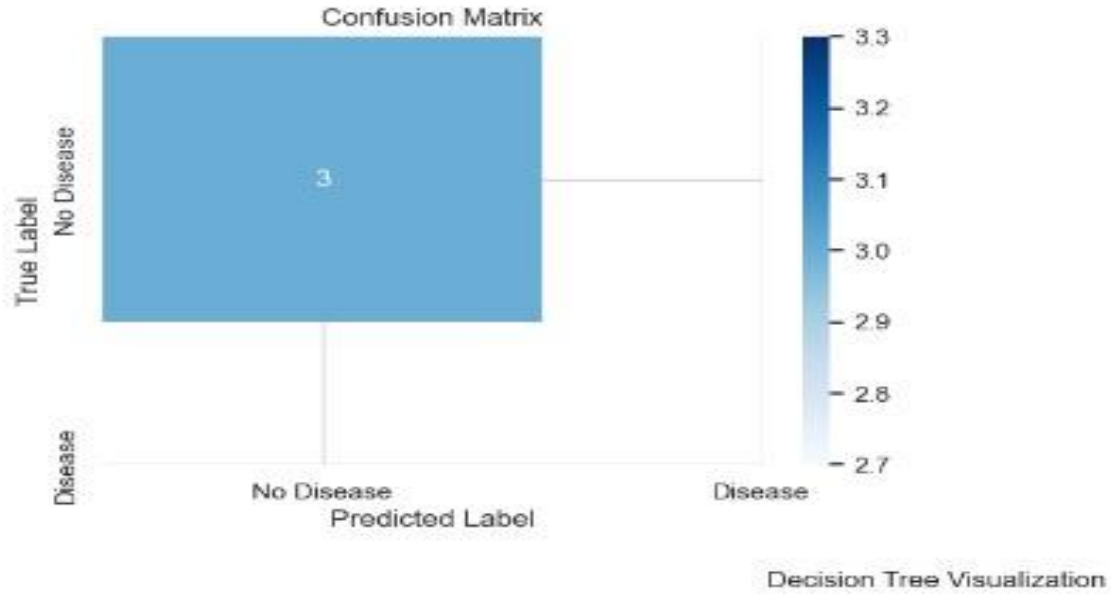
LABLE GRAPH COUNT

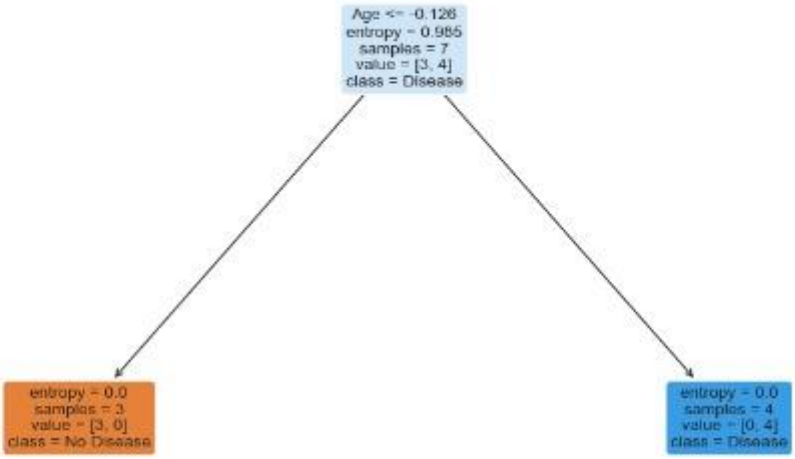


ACCURACY – CLASSIFICATION REPORT

	Age	Blood_Pressure	Cholesterol	Diabetes	Disease
0	25	120	180	0	No
1	30	130	200	0	No
2	45	140	250	1	Yes
3	35	135	220	0	No
4	50	145	230	1	Yes
5	23	118	175	0	No
6	43	138	245	1	Yes
7	55	150	260	1	Yes
8	37	136	215	0	No
9	29	125	190	0	No

PREDICTED LABLE






```
Decision Tree Rules:

|--- Age <= -0.13
|   |--- class: 0
|--- Age >  -0.13
|   |--- class: 1
```

Execution Steps:

1. Import Libraries: Import necessary libraries for data manipulation, visualization, and machine learning.

2. Set Seaborn Style: Set the aesthetic style of the plots using Seaborn.

3. Create Sample Datase: Define a sample healthcare dataset with features such as age, blood pressure, cholesterol, diabetes status, and disease presence.

4. Load Data into DataFrame: Convert the sample data dictionary into a Pandas DataFrame and display it.

5. Convert Categorical Column: Use `LabelEncoder` to convert the categorical 'Disease' column into a numerical format (0 for 'No', 1 for 'Yes').

6. Data Visualization: Create a count plot to visualize the occurrence of disease in the dataset.

7. Split Dataset: Split the dataset into features (`X`) and target variable (`y`), then divide it into training and testing sets (70% train, 30% test).

8. Feature Scaling: Apply `StandardScaler` to standardize the feature values for better model performance.

9. Build Decision Tree Classifier: Instantiate a `DecisionTreeClassifier` with specified parameters (criterion, max depth) and fit it to the training data.

10. Make Predictions: Use the trained model to make predictions on the test dataset.

11. Evaluate the Model: Calculate accuracy, confusion matrix, and classification report to assess the model's performance.

12. Visualize Confusion Matrix: Create a heatmap to visualize the confusion matrix.

13. Plot Decision Tree: Visualize the structure of the decision tree using the `plot_tree` function.

14. Display Decision Tree Rules: Export and print the rules derived from the decision tree in a text format.

These steps give a concise overview of how the code processes the dataset and builds a decision tree model for disease prediction.

Conclusion:

The application of Decision Tree classification in predicting disease occurrence from patient medical records has shown several strengths:

- **Interpretability**: Decision Trees provide a clear and interpretable model, making it easier for healthcare professionals to understand the factors contributing to disease prediction. This interpretability is crucial in healthcare,

where understanding the reasoning behind predictions can facilitate better decision-making.

- **Performance**: Depending on the dataset's quality and the chosen features, Decision Trees can achieve high accuracy rates in predicting disease occurrence. However, the performance can vary based on the complexity of the disease and the richness of the data.
- **Limitations**: While Decision Trees are effective, they are prone to overfitting, especially with noisy data or when they are too deep. Techniques like pruning or ensemble methods (e.g., Random Forests) can mitigate this issue.
- **Practical Implications**: The findings from such a model can aid in early disease detection, allowing for timely interventions and personalized patient care. By identifying high-risk patients based on their medical records, healthcare providers can allocate resources more effectively and improve health outcomes.

In summary, Decision Tree classification is a valuable tool in the healthcare domain for predicting disease occurrence from patient medical records. Its interpretability and ability to handle various data types make it suitable for this application, though careful consideration is needed to avoid overfitting and to validate the model's predictions in real-world scenarios.