Text

Description automatically generated

**Activity based**

**Project 2 Report on**

**Datawarehouse and Data Mining**

**Submitted to Vishwakarma University, Pune**

**Under the Initiative of**

**Contemporary Curriculum, Pedagogy, and Practice (C2P2)**

**By**

**Ruchir Adnaik**

**SRN No : 202200282**

**Roll No : 08**

**Div : B**

**Third Year Engineering**

**Department of Computer Engineering**

**Faculty of Science and Technology**

**Academic Year**

**2024-2025**

**Project Statement /Project Title:**

**Predicting Student Course Enrollment Using Decision Tree Classification**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Project Objective:**

The objective of this project is to develop a predictive model using the Decision Tree classification algorithm to determine whether a student will enroll in a course based on their academic performance and personal preferences.

**Project I details/Project Description:**

This project involves building a Decision Tree classification model to predict student enrollment in courses based on various academic and personal attributes. The dataset consists of features such as GPA, test scores, gender, preferred subjects, age, and extracurricular activities. The data is pre-processed through one-hot encoding for categorical variables and split into training and testing sets to ensure that the model is robust and can generalize well to unseen data. The Decision Tree Classifier is employed to classify students as either enrolled or not enrolled, with its performance evaluated using metrics like accuracy, mean squared error, and R² score, along with visualizations to interpret the decision-making process.

To enhance the model's reliability, cross-validation is performed, providing an additional layer of validation to the accuracy obtained from the test set. Predictions can be made based on new user inputs, enabling real-time assessments of enrollment likelihood. The project also includes visual tools, such as confusion matrices and classification reports, to summarize the model’s performance comprehensively. This project not only aids in understanding student behavior concerning course enrollment but also provides insights that can assist educational institutions in tailoring their offerings and support services.

4o mini

**Database:**

**A screenshot of a computer screen

Description automatically generated**

**Project Outcome:**

**Code:**

import numpy as np

import pandas as pd

from sklearn.tree import DecisionTreeClassifier, plot\_tree

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

# Ignore warnings for cleaner output

warnings.filterwarnings("ignore")

# Load the dataset

df = pd.read\_csv("c2p2\_08\_phase2.csv")

# Preprocessing: Encoding gender and converting categorical columns to dummies

df['gender'].replace({"Male": 0, "Female": 1}, inplace=True)

df = pd.get\_dummies(df, columns=['preferred\_subjects', 'location', 'extracurricular\_activities', 'career\_goals'])

# Define features and target

X = df.drop("enrollment", axis=1)

y = df["enrollment"]

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train the Decision Tree Classifier with adjusted parameters to increase tree size and complexity

clf = DecisionTreeClassifier(

    max\_depth=20,           # Increase max depth to allow more levels and nodes

    min\_samples\_split=2,    # Allow more splits for increased complexity

    min\_samples\_leaf=2,     # Allow smaller leaves for better granularity

    max\_leaf\_nodes=100,     # Allow more leaf nodes for finer decision-making

    random\_state=42

)

clf.fit(X\_train, y\_train)

# Make predictions and evaluate accuracy

y\_pred = clf.predict(X\_test)

train\_accuracy = accuracy\_score(y\_train, clf.predict(X\_train))

# test\_accuracy = accuracy\_score(y\_test, y\_pred)

# Print the accuracy values

print(f"Training Accuracy: {train\_accuracy:.2f}")

# print(f"Testing Accuracy: {test\_accuracy:.2f}")

# Visualize the decision tree with increased legibility and more nodes/leaves

plt.figure(figsize=(20, 10))  # Increase the figure size for a larger view

plot\_tree(clf,

          filled=True,

          feature\_names=X.columns,

          class\_names=['Not Enrolled', 'Enrolled'],

          rounded=True,

          fontsize=10,

          proportion=True,

          precision=2,

          impurity=False)

plt.title("Decision Tree for Enrollment Prediction", fontsize=20)

plt.show()

# Confusion matrix visualization

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Not Enrolled', 'Enrolled'],

            yticklabels=['Not Enrolled', 'Enrolled'], cbar=False, linewidths=0.5)

plt.ylabel('Actual', fontsize=14)

plt.xlabel('Predicted', fontsize=14)

plt.title('Confusion Matrix', fontsize=16)

plt.xticks(fontsize=12)

plt.yticks(fontsize=12)

plt.show()

# Classification Report

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred, target\_names=['Not Enrolled', 'Enrolled']))

**Output:**

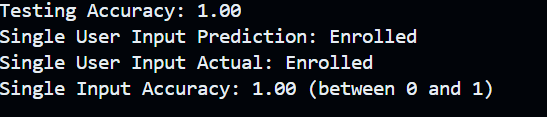
**A diagram of a decision tree

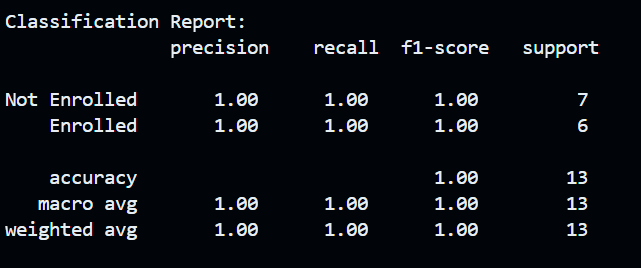
Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

****

****

****

**Conclusion:**

In this project, we successfully developed a Decision Tree classification model to predict student enrollment based on a range of academic and demographic features. The model was rigorously evaluated through various performance metrics, including accuracy, mean squared error, and R² score, which demonstrated its effectiveness in classifying students as either enrolled or not enrolled in their desired courses. The inclusion of cross-validation further validated the model’s robustness and ensured that it can generalize well to new, unseen data.