DEPARTMENT OF COMPUTER ENGINEERING

Semester	T.E. Semester V – Computer Engineering
Subject	Data Warehousing and Mining
Subject Professor In-charge	Prof. Kavita P Shirsat
Laboratory	Lab 313-A

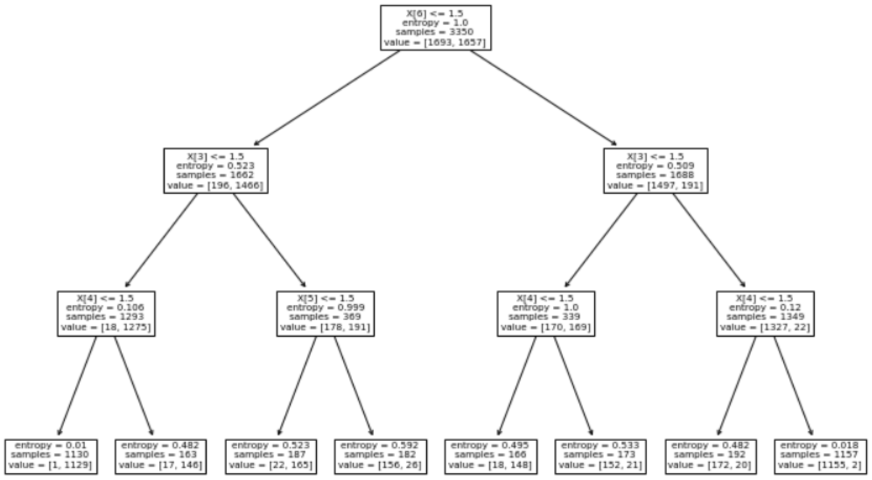
Student Name	Aditya Oza	
Roll Number	20102A0031	
Grade and Subject Teacher's Signature		

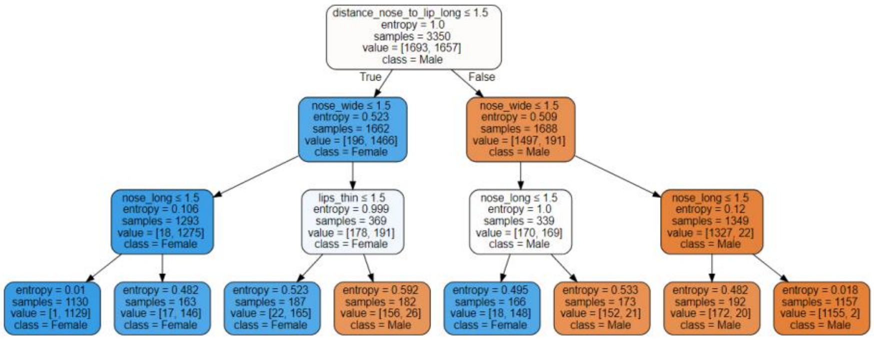
Experiment Title	PBL – Implementation of Decision Tree Algorithm	
Resources / Apparatus Required	Hardware: Computer system	Software: Python
Objectives (Skill Set / Knowledge Tested / Imparted)	Implementation of Decision Tree on real life dataset	

Code

```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt # data visualization

import seaborn as sns # statistical data visualization

import warnings

warnings.filterwarnings('ignore')

from google.colab import files

uploaded = files.upload()



df = pd.read_csv("gender_classification_v7.csv")



df.head()



col_names = ['long_hair', 'forehead_width_cm', 'forehead_height_cm', 'nose_wide',
'nose_long', 'lips_thin', 'distance_nose_to_lip_long','gender']

df.columns = col_names

col_names

df.head()

col_names = ['long_hair', 'forehead_width_cm', 'forehead_height_cm', 'nose_wide', 'nose_long', 'lips_thin', 'distance_nose_to_lip_long','gender']

for col in col_names:

 print(df[col].value_counts())

df['gender'].value_counts()

# check missing values in variables

df.isnull().sum()

X = df.drop(['gender'], axis=1)

y = df['gender']

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)

# check the shape of X_train and X_test

X_train.shape, X_test.shape

# check data types in X_train

X_train.dtypes

X_train.head()

!pip install category_encoders

# import category encoders

import category_encoders as ce

# encode variables with ordinal encoding

encoder = ce.OrdinalEncoder(cols=['long_hair', 'forehead_width_cm', 'forehead_height_cm', 'nose_wide', 'nose_long', 'lips_thin', 'distance_nose_to_lip_long'])
```

```
X_train = encoder.fit_transform(X_train)
X_test = encoder.transform(X_test)
X_train.head()
X_test.head()
# import DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier
# instantiate the DecisionTreeClassifier model with criterion entropy
clf_en = DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)
# fit the model
clf_en.fit(X_train, y_train)
y_pred_en = clf_en.predict(X_test)
from sklearn.metrics import accuracy_score
print('Model accuracy score with criterion entropy: {0:0.4f}'.
format(accuracy_score(y_test, y_pred_en)))
y_pred_train_en = clf_en.predict(X_train)
y_pred_train_en
print('Training-set accuracy score: {0:0.4f}'. format(accuracy_score(y_train,
y_pred_train_en)))
# print the scores on training and test set
print('Training set score: {:.4f}'.format(clf_en.score(X_train, y_train)))
print('Test set score: {:.4f}'.format(clf_en.score(X_test, y_test)))
plt.figure(figsize=(12,8))
from sklearn import tree
tree.plot_tree(clf_en.fit(X_train, y_train))


import graphviz
dot_data = tree.export_graphviz(clf_en, out_file=None,
                    feature_names=X_train.columns,
                              class_names=y_train,
                              filled=True, rounded=True,
                              special_characters=True)
graph = graphviz.Source(dot_data)
graph
```





Coclusion	We have successfully implemented decision tree on real life dataset.