Laporan Praktikum
Pembelajaran Mesin

Disusun oleh :
Kelompok 3

1. Gede Rangga Wira Aditya (082011633048)
2. Muhammad Rahmadhani Ferdiansyah (082011633068)
3. Arya Danu Triatmodjo (082011633069)
4. Mukhamad Ikhsanudin (082011633086)

Kelas I3

PROGRAM STUDI S1 SISTEM INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS AIRLANGGA

SURABAYA

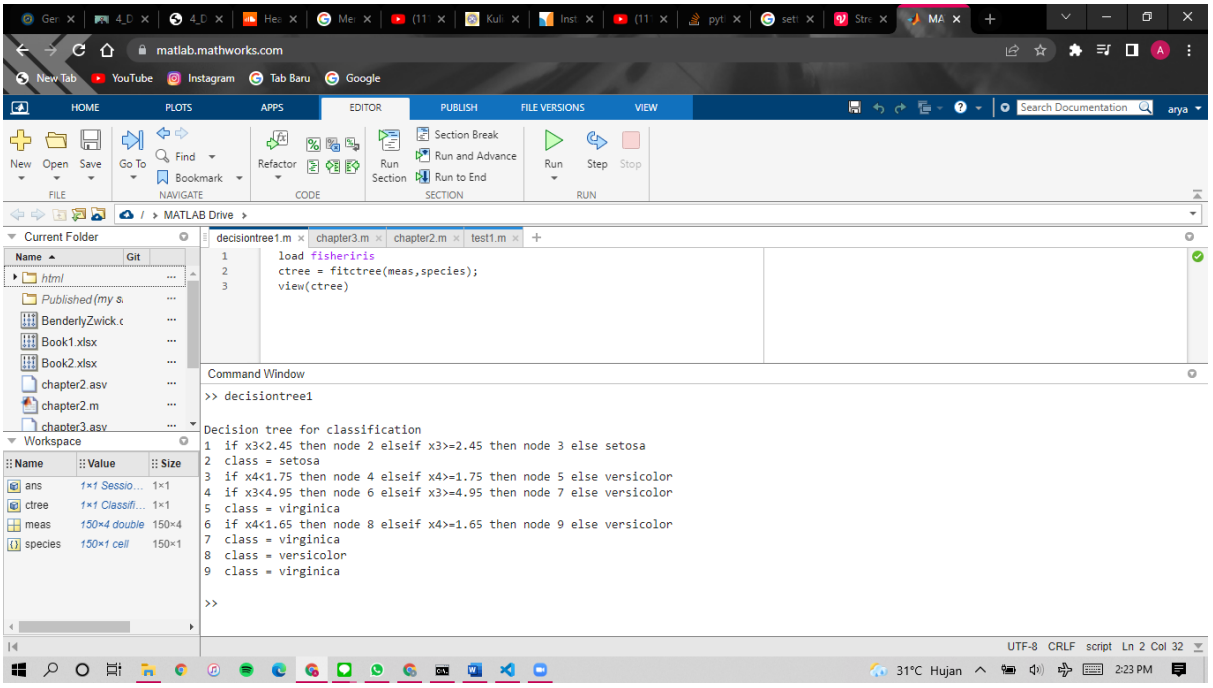
2021/2022

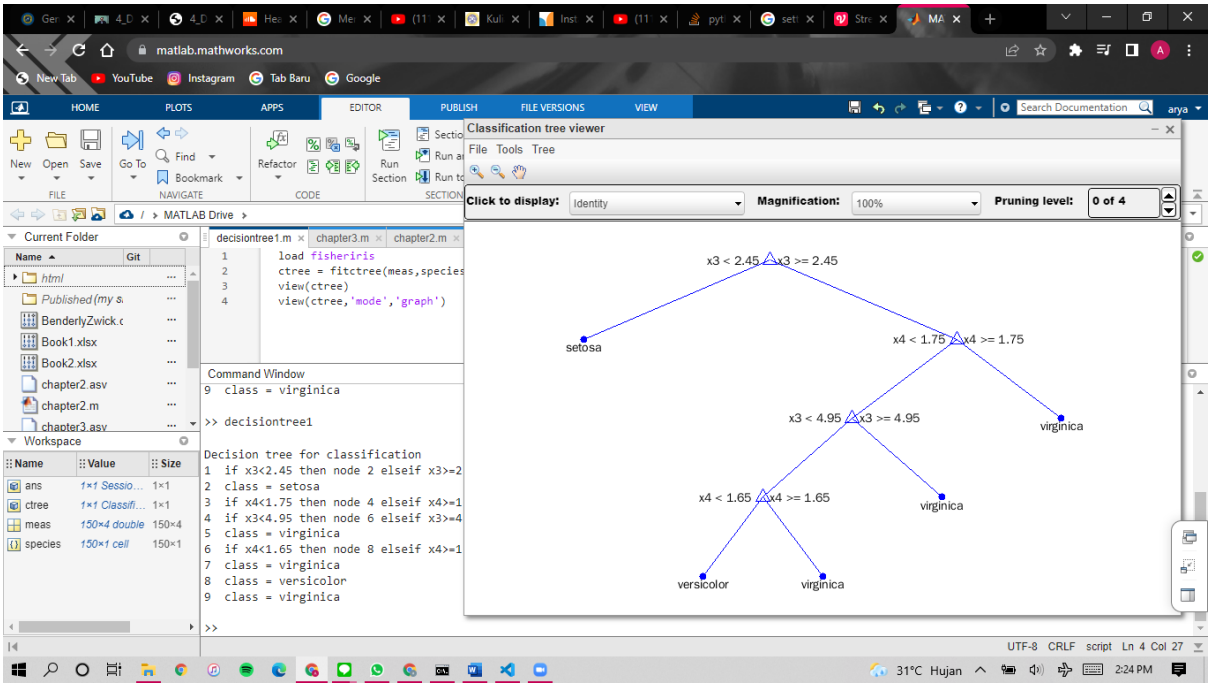
Decision Tree, Decision trees, or classification trees and regression trees, predict responses to data. To predict a response, follow the decisions in the tree from the root (beginning) node down to a leaf node. The leaf node contains the response. This example shows how to view a classification or regression tree. There are two ways to view a tree: view(tree) returns a text description and view(tree,'mode','graph') returns a graphic description of the tree.

MATLAB

Try all the source code

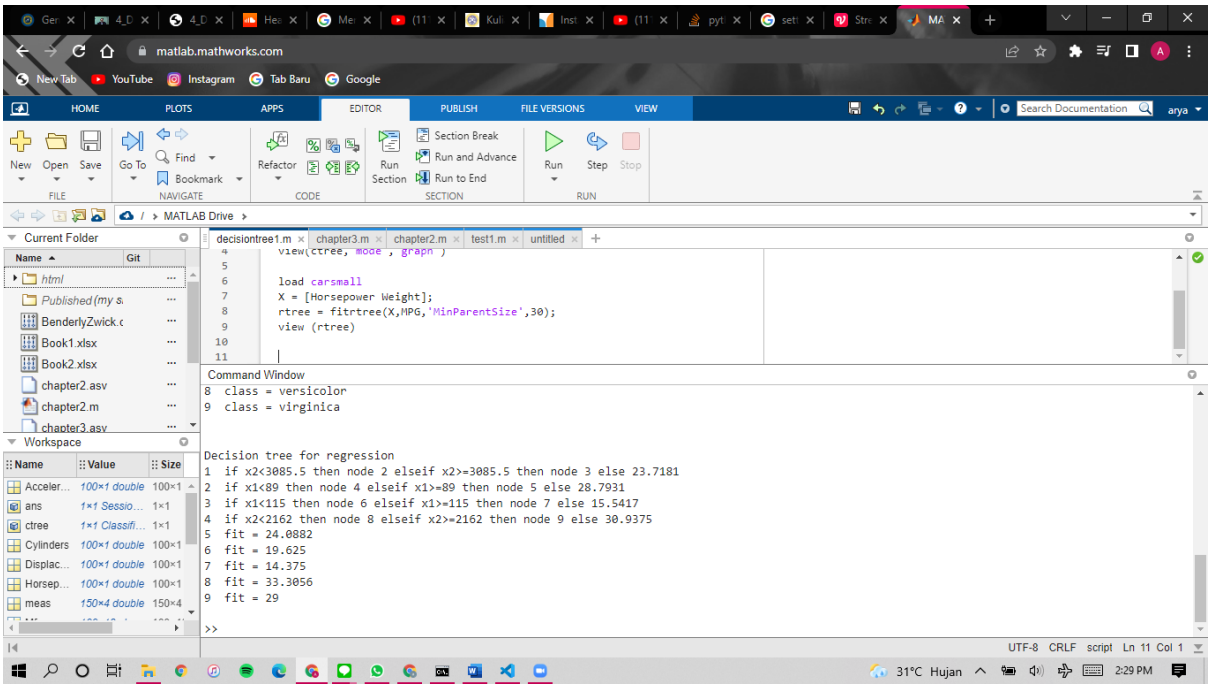
Example 1:
Create and view a classification tree.
Result :

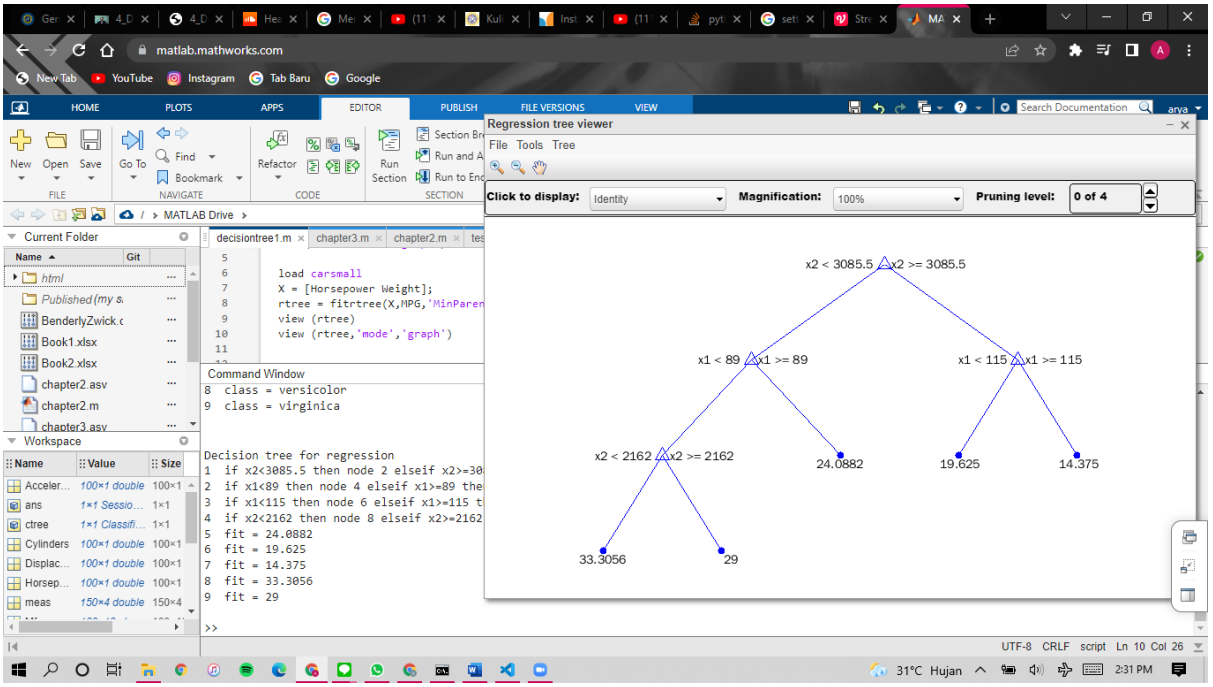




Now, create and view a regression tree.

Result :

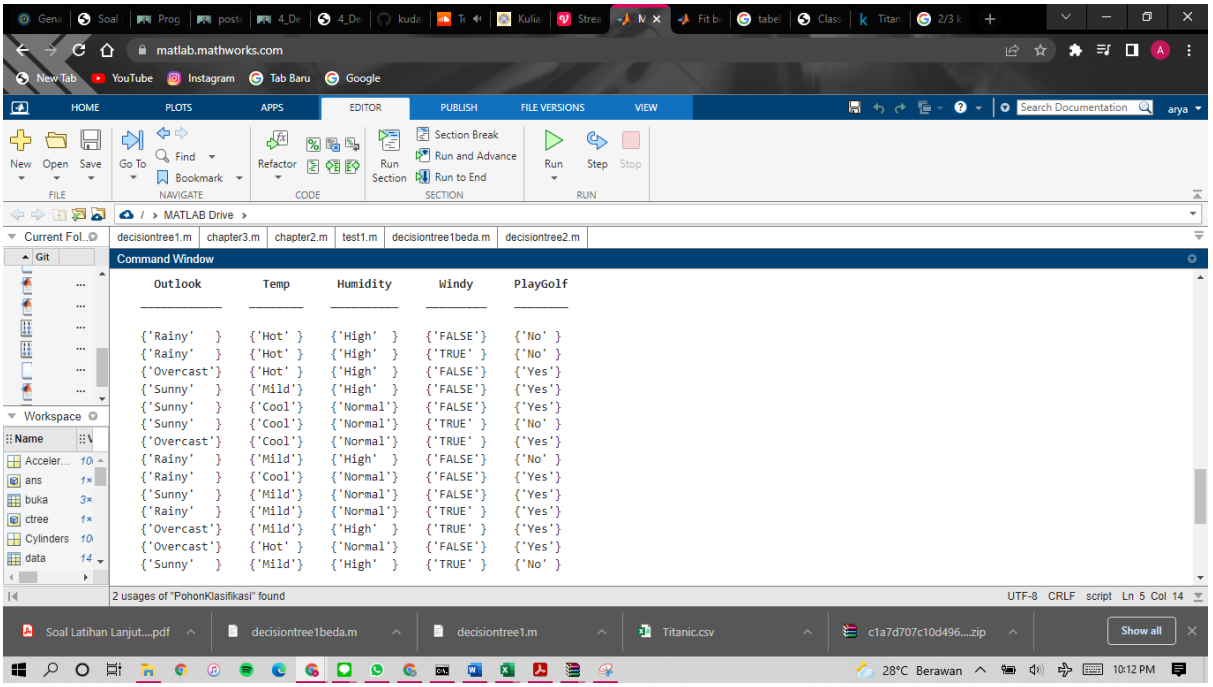




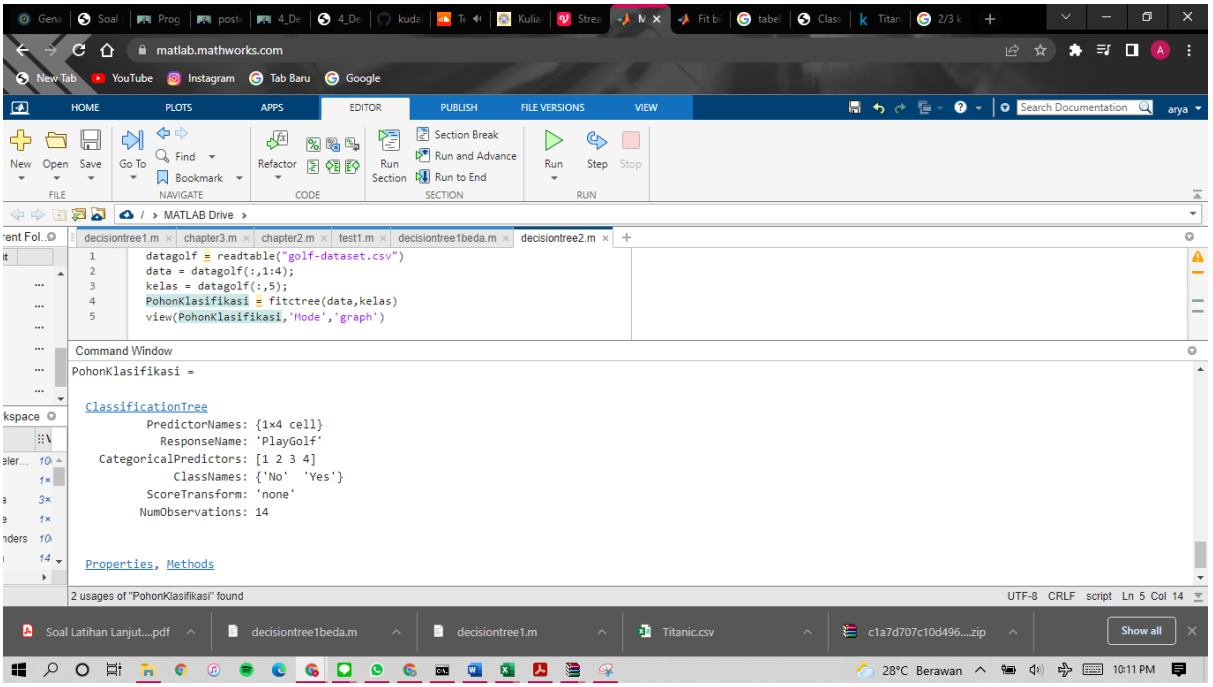
Example 2:

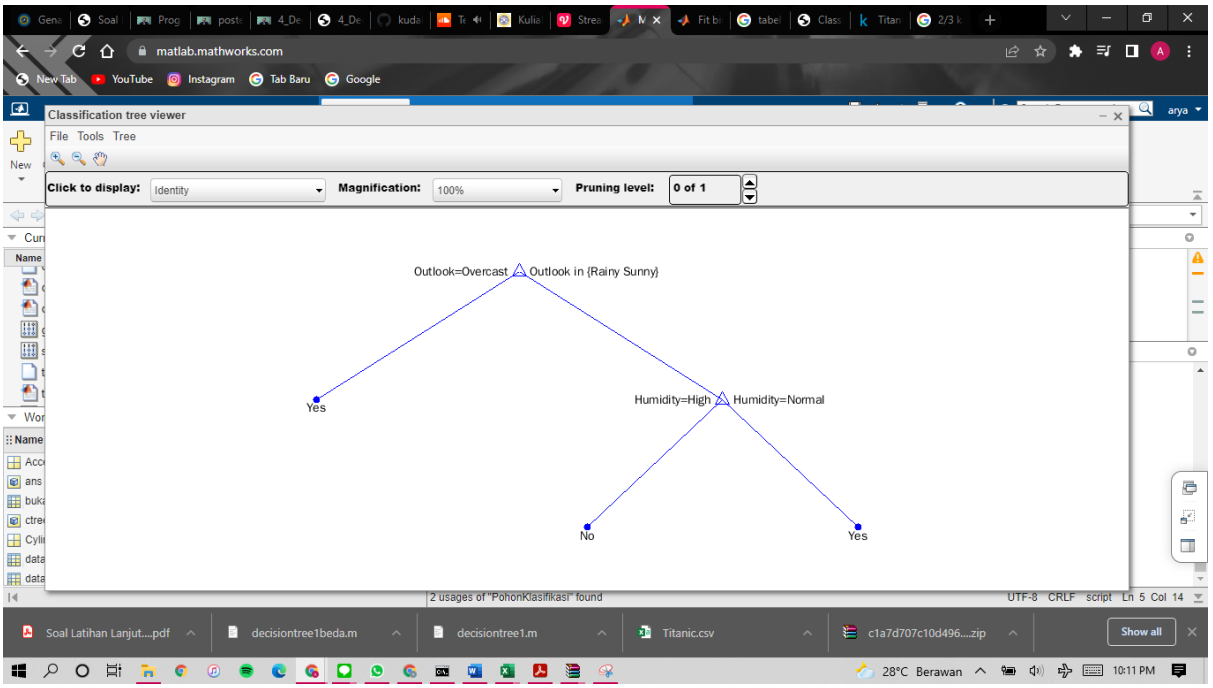
Source Code : https://gist.github.com/kudaliar032/b8cf65d84b73903257ed603f6c1a2508

Data :



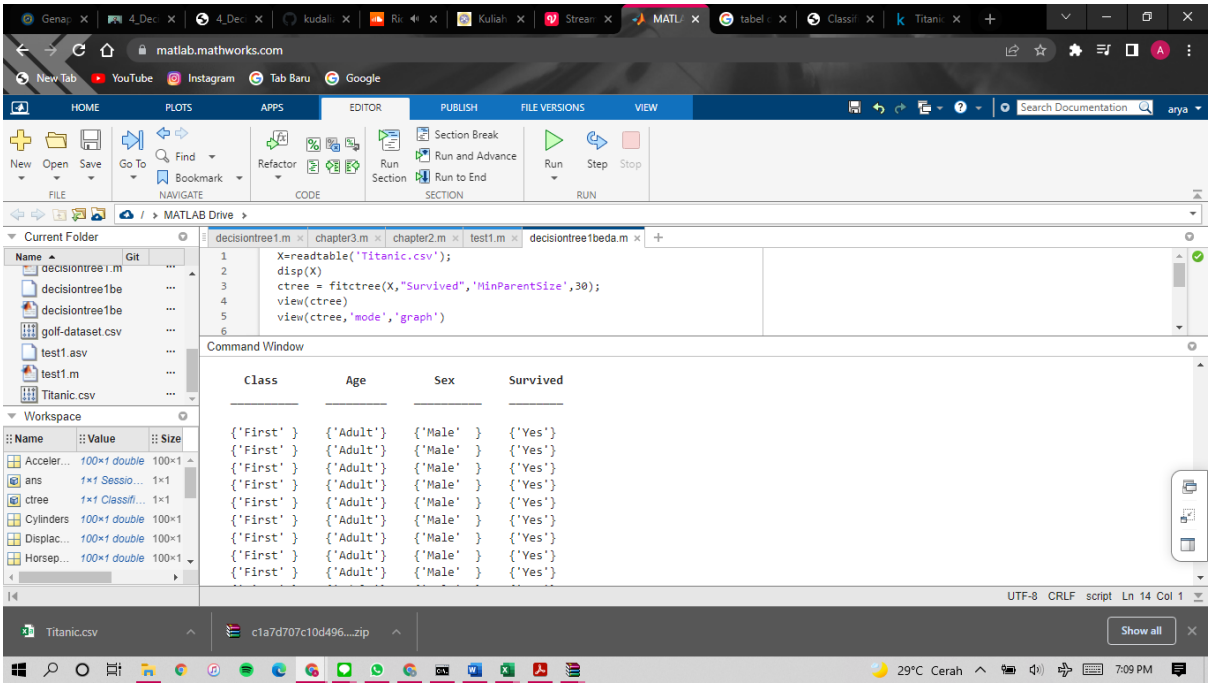
Result :



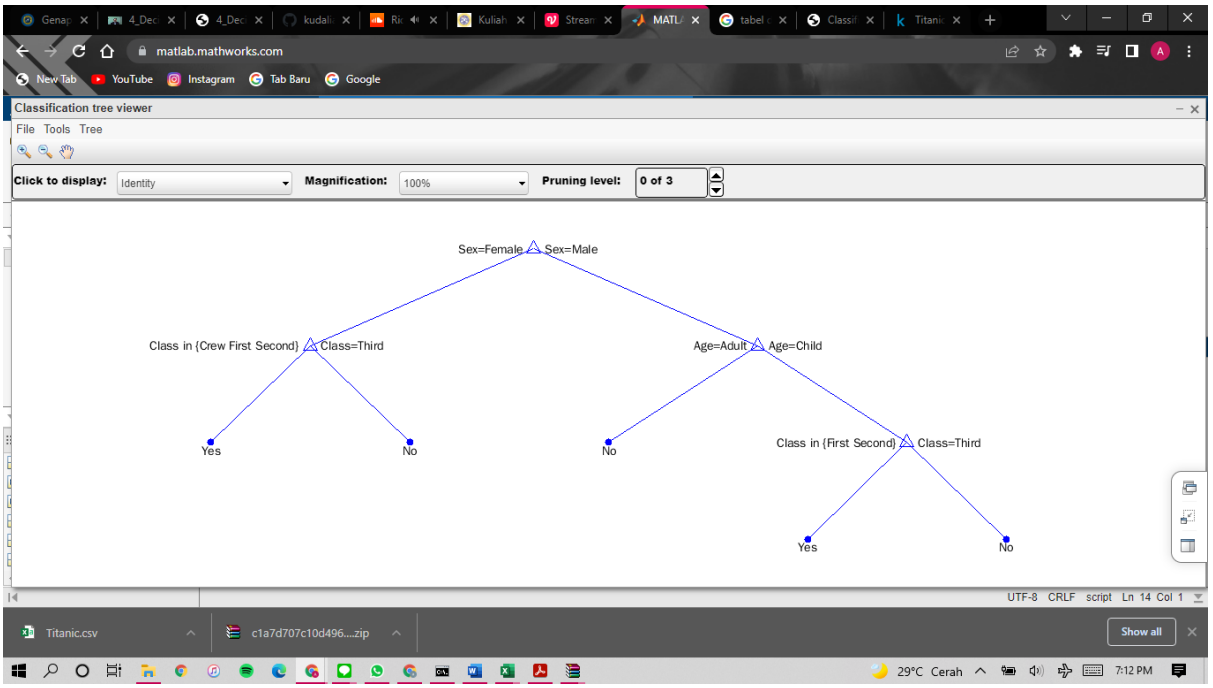


Using other data

Example 1:
Create and view a classification tree.
Result :

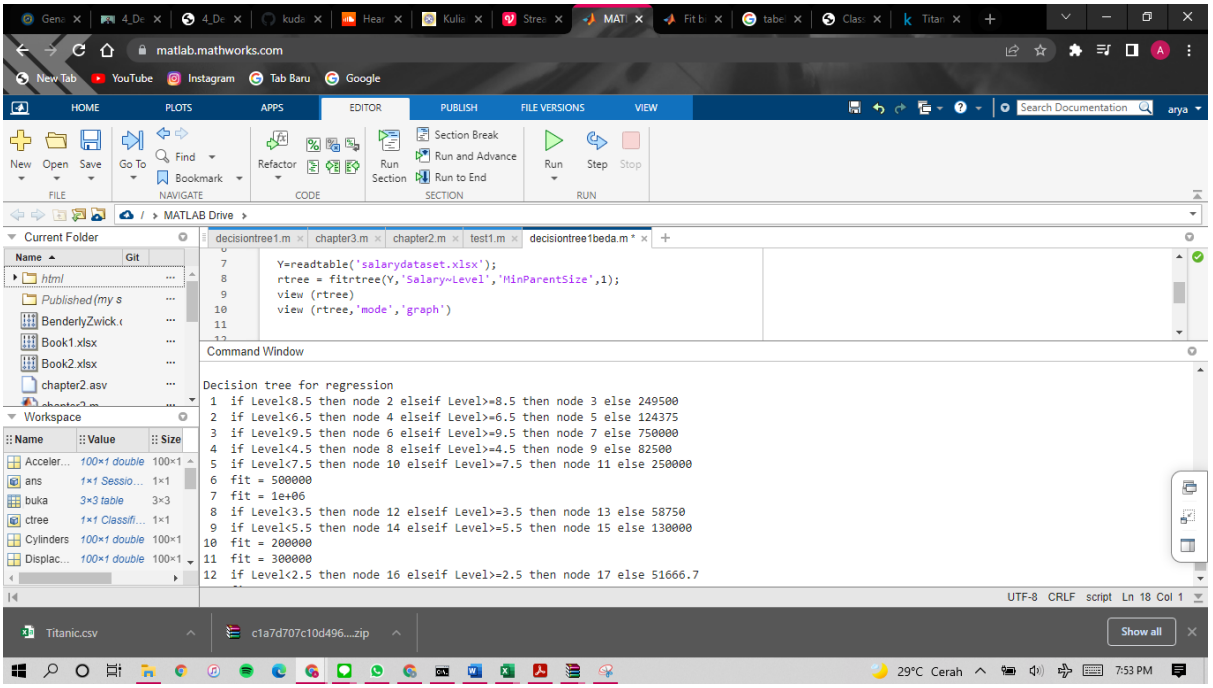


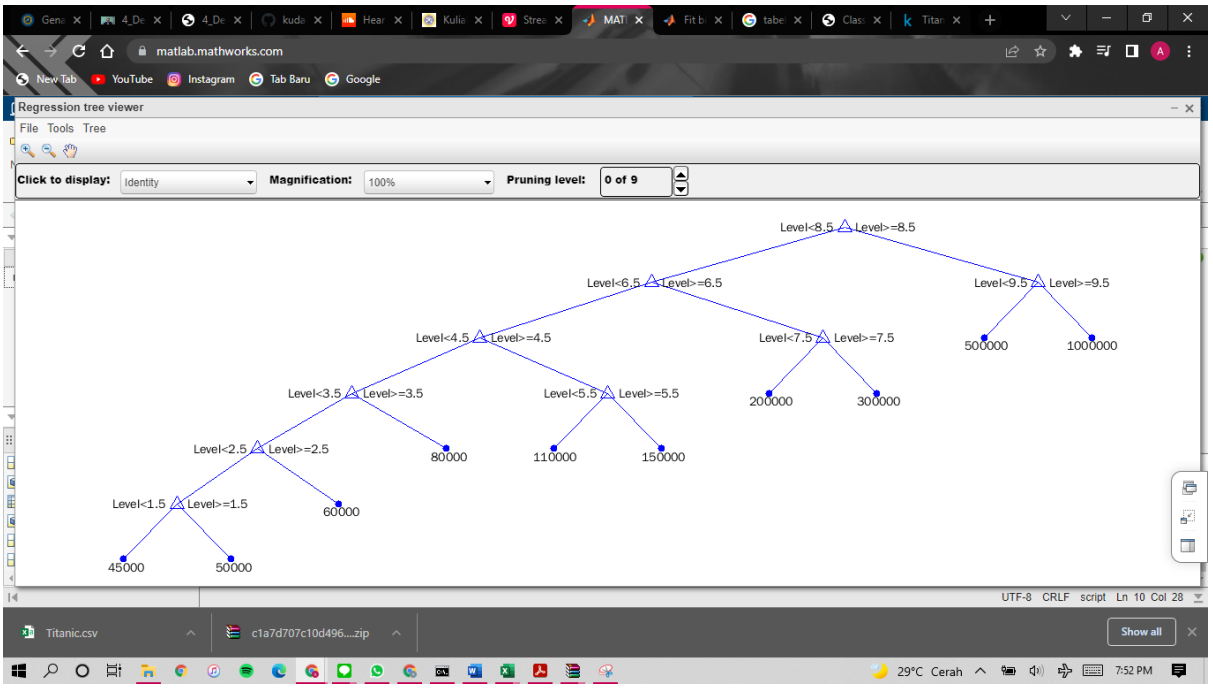

```
Decision tree for classification
1  if Sex=Female then node 2 elseif Sex=Male then node 3 else No
2  if Class in {Crew First Second} then node 4 elseif Class=Third then node 5 else Yes
3  if Age=Adult then node 6 elseif Age=Child then node 7 else No
4  class = Yes
5  class = No
6  class = No
7  if Class in {First Second} then node 8 elseif Class=Third then node 9 else No
8  class = Yes
9  class = No
```



Now, create and view a regression tree.

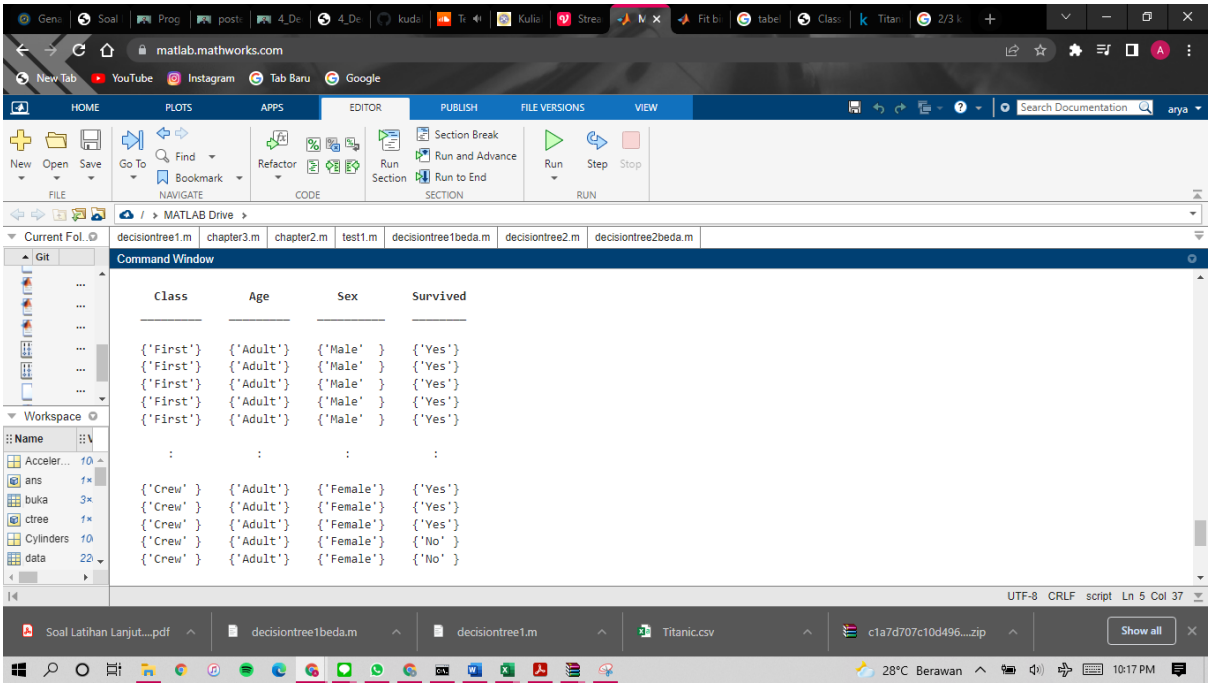
Result :



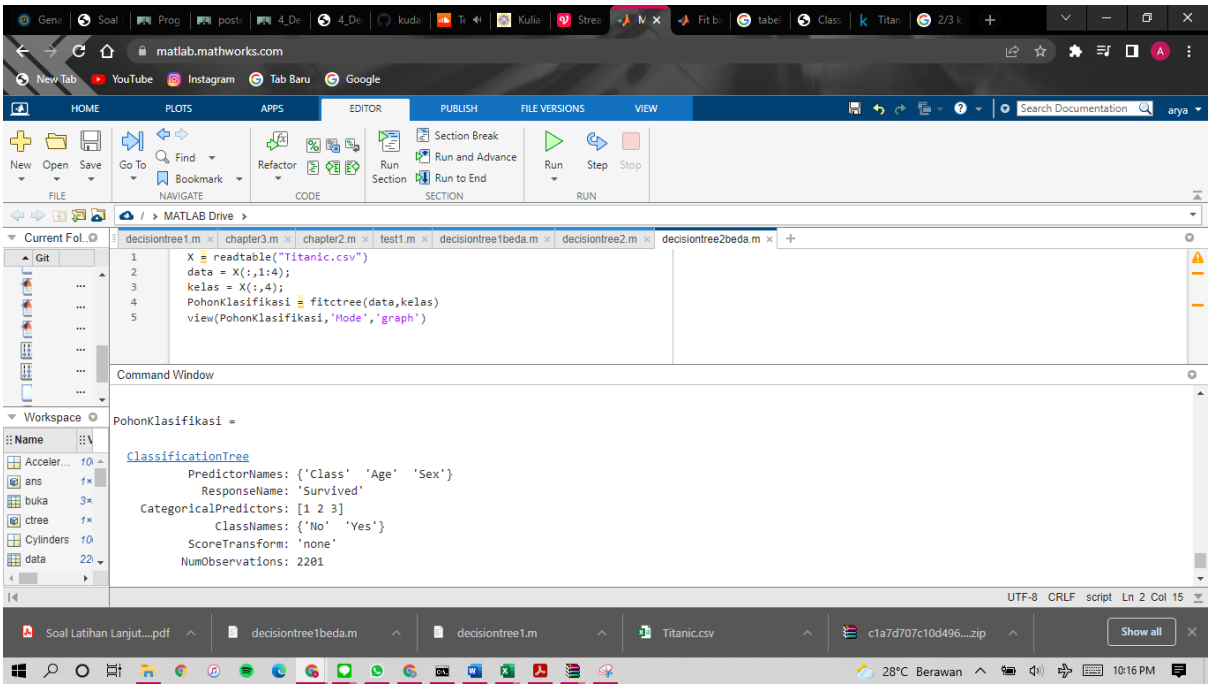


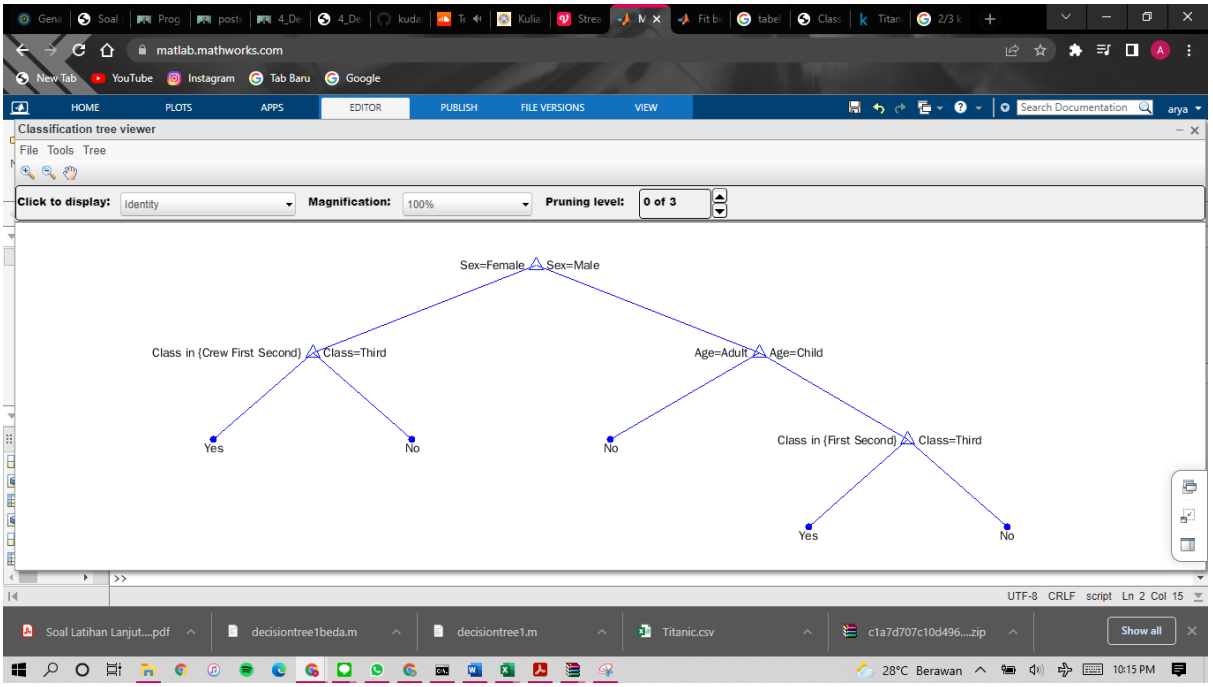
Example 2:

Data :


matlab.mathworks.com
HOME PLOTS APPS EDITOR PUBLISH FILE VERSIONS VIEW
MATLAB Drive
decisiontree1.m chapter3.m chapter2.m test1.m decisiontree1beda.m decisiontree2.m decisiontree2beda.m
Command Window
Class Age Sex Survived
{'First'} {'Adult'} {'Male'} {'Yes'}
{'First'} {'Adult'} {'Male'} {'Yes'}
{'First'} {'Adult'} {'Male'} {'Yes'}
{'First'} {'Adult'} {'Male'} {'Yes'}
{'First'} {'Adult'} {'Male'} {'Yes'}
{'Crew'} {'Adult'} {'Female'} {'Yes'}
{'Crew'} {'Adult'} {'Female'} {'Yes'}
{'Crew'} {'Adult'} {'Female'} {'Yes'}
{'Crew'} {'Adult'} {'Female'} {'No'}
{'Crew'} {'Adult'} {'Female'} {'No'}
Workspace
Soal Latihan Lanjut....pdf
decisiontree1beda.m
decisiontree1.m
Titanic.csv
c1a7d707c10d496....zip
Show all
28°C Berawan
10:17 PM

Result :


matlab.mathworks.com
HOME PLOTS APPS EDITOR PUBLISH FILE VERSIONS VIEW
MATLAB Drive
decisiontree1.m chapter3.m chapter2.m test1.m decisiontree1beda.m decisiontree2.m decisiontree2beda.m
X = readtable("Titanic.csv")
data = X(:,1:4);
kelas = X(:,4);
PohonKlasifikasi = fitctree(data,kelas)
view(PohonKlasifikasi,'Mode','graph')
Command Window
PohonKlasifikasi =
ClassificationTree
PredictorNames: {'Class' 'Age' 'Sex'}
ResponseName: 'Survived'
CategoricalPredictors: [1 2 3]
ClassNames: {'No' 'Yes'}
ScoreTransform: 'none'
NumObservations: 2201
Workspace
Soal Latihan Lanjut....pdf
decisiontree1beda.m
decisiontree1.m
Titanic.csv
c1a7d707c10d496....zip
Show all
28°C Berawan
10:16 PM



PYTHON

Try all the source code

Importing Required Libraries

First, import the required modules, and read the dataset with pandas.

```
import pandas
from sklearn import tree
import pydotplus
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
import matplotlib.image as pltimg
```

Loading Data and check is there any missing value

Let's first load the required Titanic dataset using pandas' read CSV function.

```
df = pandas.read_csv(r"C:\Users\ACER\Downloads\Titanic.csv")
df.info()
```

Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2201 entries, 0 to 2200
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Class     2201 non-null   object
 1   Age       2201 non-null   object
 2   Sex       2201 non-null   object
 3   Survived  2201 non-null   object
dtypes: object(4)
memory usage: 68.9+ KB
```

convert value from char/string to numeric

To create a decision tree, all data must be numeric. We have to convert the non numeric columns 'Class' and 'Sex','Age','Survived' into numeric values.

```
10  d = {'First': 0, 'Second': 1, 'Crew': 2, 'Third':3}
11  df['Class'] = df['Class'].map(d)
12  d = {'Adult': 1, 'Child': 2}
13  df['Age'] = df['Age'].map(d)
14  d = {'Female': 1, 'Male': 2}
15  df['Sex'] = df['Sex'].map(d)
16  d = {'Yes': 1, 'No': 0}
17  df['Survived'] = df['Survived'].map(d)
18
```

Feature Selection

Then we have to separate the feature column from the target column. The feature column is the column we are trying to predict, and the target column is the column with the values we are trying to predict. Example: X is the feature column, y is the target column

```
19  features = ['Class', 'Age', 'Sex']
20
21  X = df[features]
22  y = df['Survived']
23
24  print(X)
25  print(y)
```

Output

```
      Class  Age  Sex
0         0    1    2
1         0    1    2
2         0    1    2
3         0    1    2
4         0    1    2
...     ...  ...  ...
2196      2    1    1
2197      2    1    1
2198      2    1    1
2199      2    1    1
2200      2    1    1

[2201 rows x 3 columns]
0       1
1       1
2       1
3       1
4       1
       ..
2196    1
2197    1
2198    1
2199    0
2200    0
Name: Survived, Length: 2201, dtype: int64
```

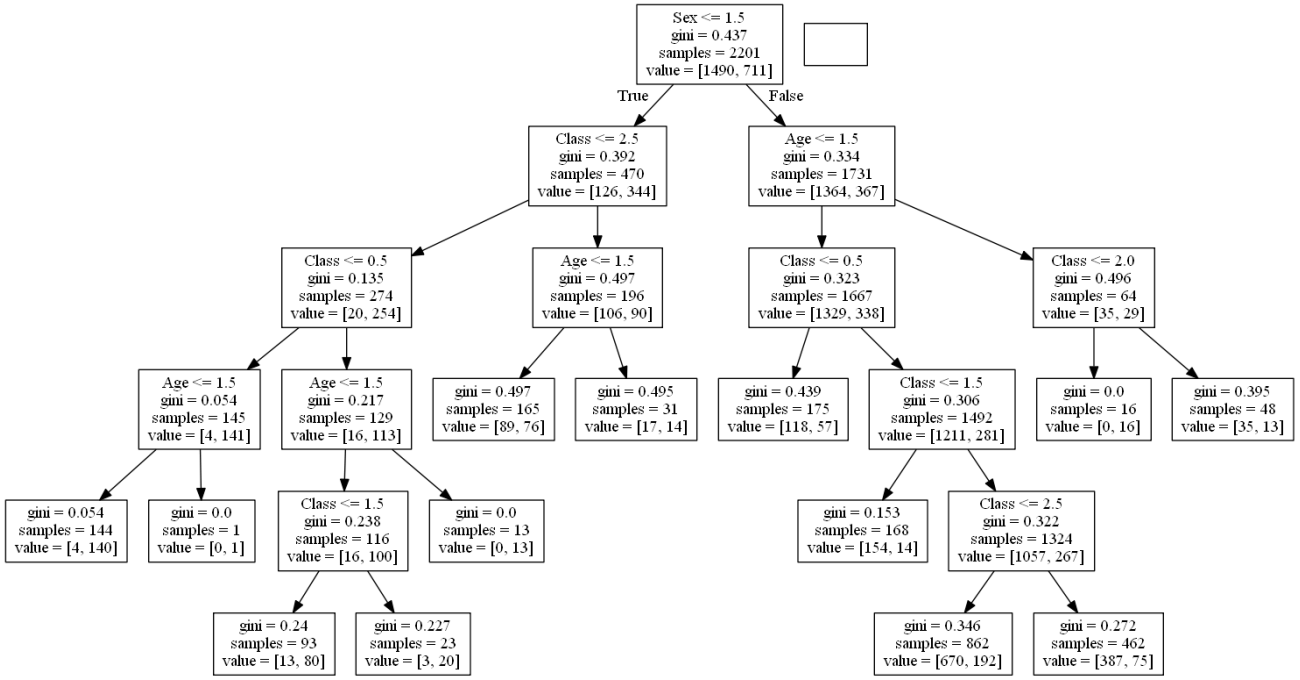
Visualizing Decision Trees

Now we can create the actual decision tree, adapt it to those details, and save the .png file on the computer.

```
dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)
data = tree.export_graphviz(dtree, out_file=None, feature_names=features)
graph = pydotplus.graph_from_dot_data(data)
graph.write_png('mydecisiontree.png')

img=pltimg.imread('mydecisiontree.png')
imgplot = plt.imshow(img)
plt.show()
```

Output



The other method

a. **Import the library that needed**

```
import pandas as pd
from sklearn import preprocessing, tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
```

b. **Read the data and check the information of missing value**

```
# READ DATA & CHECK THE VARIABLE
data = pd.read_csv("gender_classification_v7.csv")
df = pd.DataFrame(data)
df.info()
```

c. **Make one of variable to be the target variable**

```
# TARGET VARIABLE
gender = df['gender']
```

d. **Convert the target variable data type to numeric or binary**

```
# CONVERT THE TARGET VALUE INTO NUMERIC / BINARY
d = {"Male":1, "Female":0}
df['gender'] = df['gender'].map(d)
```

e. **Make the condition for the target variable with other variable**

```
# MAKE THE CONDITION INTO FEATURES FOR THE TARGET VARIABLE
features = ["long_hair", "forehead_width_cm", "forehead_height_cm",
            "nose_wide", "nose_long", "lips_thin", "distance_nose_to_lip_long"]
```

f. **We can make the variable simpler to check**

```
# MAKE THE TARGET AND CONDITION SIMPLER
X = df[features]
Y = df['gender']
```

g. **Classificate the variable**

```
# CLASSIFICATE THE VARIABLE
clf = DecisionTreeClassifier(max_depth = 3)
model = clf.fit(X, Y)
```

h. **The result of classification in the text**

```
# RESULT THE CLASSIFICATION IN TEXT
text_representation = tree.export_text(clf)
print(text_representation)
```

i. **The result of the classification in the image**

```
# VISUALIZATING THE CLASSIFICATION
fig = plt.figure(figsize=(25,20))
plot = tree.plot_tree(clf,
                      feature_names = features,
                      class_names = gender,
                      filled = True)

# SAVING THE IMAGE CLASSIFICATION RESULT
fig.savefig("gender_classification.png")
```

Output

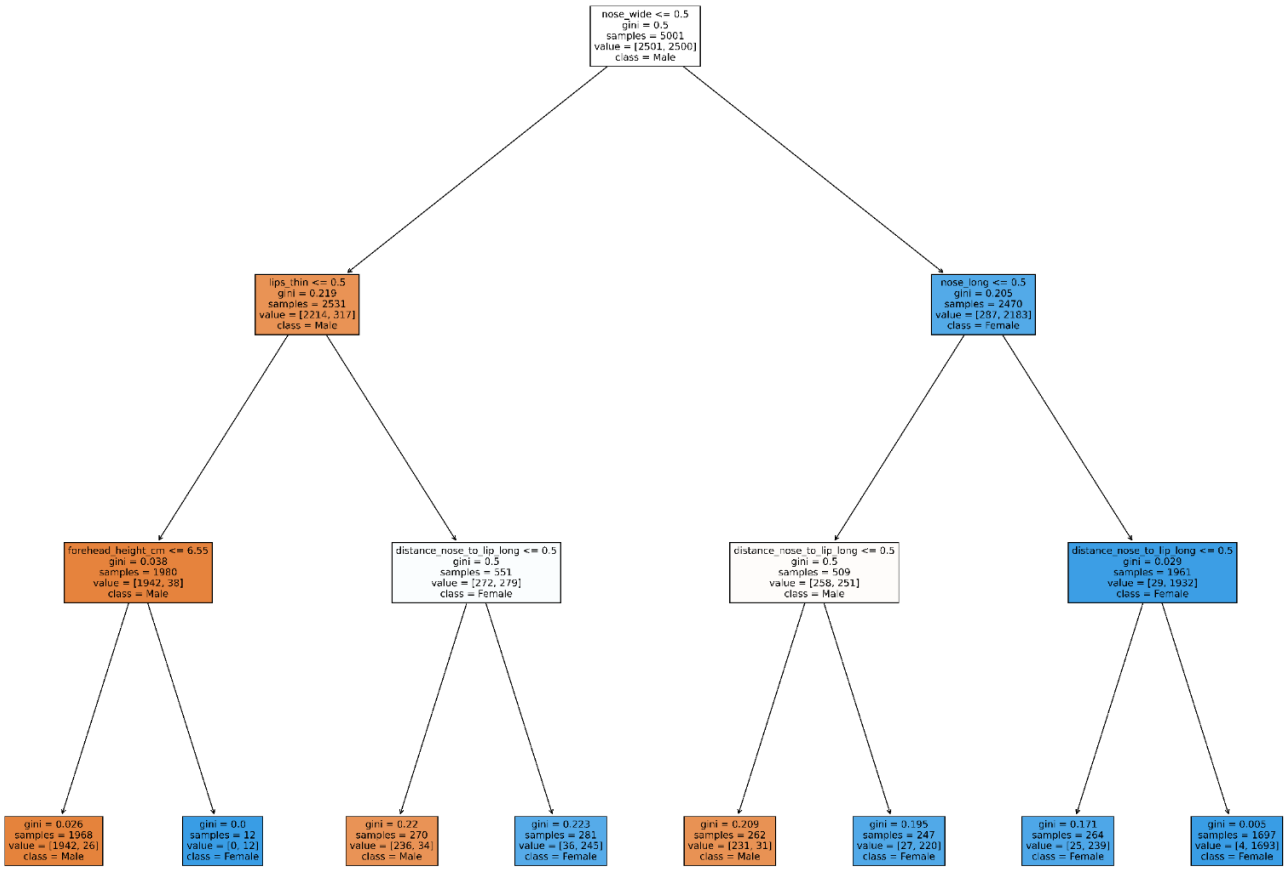
Information of the data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5001 entries, 0 to 5000
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   long_hair                  5001 non-null   int64
 1   forehead_width_cm          5001 non-null   float64
 2   forehead_height_cm         5001 non-null   float64
 3   nose_wide                  5001 non-null   int64
 4   nose_long                  5001 non-null   int64
 5   lips_thin                  5001 non-null   int64
 6   distance_nose_to_lip_long  5001 non-null   int64
 7   gender                     5001 non-null   object
dtypes: float64(2), int64(5), object(1)
```

Result in the text

```
|--- feature_3 <= 0.50
|   |--- feature_5 <= 0.50
|   |   |--- feature_2 <= 6.55
|   |   |   |--- class: 0
|   |   |--- feature_2 >  6.55
|   |   |   |--- class: 1
|   |--- feature_5 >  0.50
|   |   |--- feature_6 <= 0.50
|   |   |   |--- class: 0
|   |   |--- feature_6 >  0.50
|   |   |   |--- class: 1
|--- feature_3 >  0.50
|   |--- feature_4 <= 0.50
|   |   |--- feature_6 <= 0.50
|   |   |   |--- class: 0
|   |   |--- feature_6 >  0.50
|   |   |   |--- class: 1
|   |--- feature_4 >  0.50
|   |   |--- feature_6 <= 0.50
|   |   |   |--- class: 1
|   |   |--- feature_6 >  0.50
|   |   |   |--- class: 1
```

Result in the image


nose_wide <= 0.5
gini = 0.5
samples = 5001
value = [2501, 2500]
class = Male
lips_thin <= 0.5
gini = 0.219
samples = 2531
value = [2214, 317]
class = Male
nose_long <= 0.5
gini = 0.205
samples = 2470
value = [287, 2183]
class = Female
forehead_height_cm <= 6.55
gini = 0.038
samples = 1980
value = [1942, 38]
class = Male
distance_nose_to_lip_long <= 0.5
gini = 0.5
samples = 551
value = [272, 279]
class = Female
distance_nose_to_lip_long <= 0.5
gini = 0.5
samples = 509
value = [258, 251]
class = Male
distance_nose_to_lip_long <= 0.5
gini = 0.029
samples = 1961
value = [29, 1932]
class = Female
gini = 0.026
samples = 1968
value = [1942, 26]
class = Male
gini = 0.0
samples = 12
value = [0, 12]
class = Female
gini = 0.22
samples = 270
value = [236, 34]
class = Male
gini = 0.223
samples = 281
value = [36, 245]
class = Female
gini = 0.209
samples = 262
value = [231, 31]
class = Male
gini = 0.195
samples = 247
value = [27, 220]
class = Female
gini = 0.171
samples = 264
value = [25, 239]
class = Female
gini = 0.005
samples = 1697
value = [4, 1693]
class = Female