Baranga George – Marius

Grupa 22C31A

Proiect RN – clasificare/predictive pret casti pc

import pandas as pd

import numpy as np

import seaborn as sns

from sklearn.tree import DecisionTreeClassifier

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

from sklearn.metrics import mean\_squared\_error, r2\_score, accuracy\_score

import pandas as pd

from keras.models import Sequential

from keras.layers import Dense

# import numpy as np

# import pandas as pd

# import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

# https://machinelearningknowledge.ai/decision-tree-classifier-in-python-sklearn-with-example/

# https://thepythonguru.com/python-string-formatting/

#incarc datele

casti = pd.read\_csv('nn2.csv')

#Transform data

casti.head(74)

Brand Frecventa casti Frecventa microfon Impedanta Pret

0 1 20000 20000 39.0 mediu

1 1 20000 6500 22.0 mic

2 1 20000 8000 45.0 mediu

3 1 20000 8000 45.0 mediu

4 1 20000 8000 45.0 mare

... ... ... ... ... ...

69 5 40000 10000 35.0 mare

70 5 20000 10000 32.0 mediu

71 5 40000 8000 32.0 mare

72 6 20000 10000 32.0 mediu

73 6 40000 10000 32.0 mare

74 rows × 5 columns

print(casti.shape)

casti.info()

# sns.countplot(casti['Pret'])

# sns.countplot(casti['Frecventa casti'],hue=casti['Pret'])

sns.countplot(casti['Impedanta'],hue=casti['Pret'])

(79, 5)

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 79 entries, 0 to 78

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Brand 79 non-null int64

1 Frecventa casti 79 non-null int64

2 Frecventa microfon 79 non-null int64

3 Impedanta 79 non-null float64

4 Pret 79 non-null object

dtypes: float64(1), int64(3), object(1)

memory usage: 3.2+ KB

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f50db7c41d0>

Chart, bar chart

Description automatically generated

f=['Pret','Brand','Frecventa casti','Frecventa microfon','Impedanta']

casti1=casti[f]

#impart setul de date: prima coloana de iesire(pret), celelalte 4 de intrare

x=casti1[f[1:]]

y=casti1['Pret']

x.head(79)

Brand Frecventa casti Frecventa microfon Impedanta

0 1 20000 20000 39.0

1 1 20000 6500 22.0

2 1 20000 8000 45.0

3 1 20000 8000 45.0

4 1 20000 8000 45.0

... ... ... ... ...

74 6 40000 10000 32.0

75 6 22000 10000 32.0

76 6 40000 12000 35.0

77 6 20000 10000 32.0

78 6 20000 6500 32.0

79 rows × 4 columns

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y,test\_size=0.2)

print(x\_train.shape)

print(x\_test.shape)

print(y\_train.shape)

print(y\_test.shape)

(63, 4)

(16, 4)

(63,)

(16,)

from keras.models import Sequential

from keras.layers import Dense

from tensorflow.keras.optimizers import SGD,Adam

model = DecisionTreeClassifier()

# criterion="gini", random\_state=42,max\_depth=3, min\_samples\_leaf=5

model.fit(x\_train,y\_train)

# from sklearn.metrics import confusion\_matrix

# print('Confusion matrix ')

DecisionTreeClassifier()

p = model.predict(x\_test)

p1 = model.predict(x\_train)

rez = accuracy\_score(y\_test,p)

rez2 = accuracy\_score(y\_train,p1)

print(p)

print(p1)

print('Test accuracy: %.2f' % (rez\*100))

print('Train accuracy: %.2f' % (rez2\*100))

['mare' 'mare' 'mic' 'mediu' 'mare' 'mic' 'mediu' 'mare' 'mic' 'mare'

'mare' 'mare' 'mare' 'mic' 'mic' 'mare']

['mic' 'mediu' 'mare' 'mare' 'mediu' 'mare' 'mare' 'mic' 'mediu' 'mare'

'mare' 'mare' 'mare' 'mare' 'mic' 'mic' 'mare' 'mare' 'mare' 'mare' 'mic'

'mare' 'mare' 'mare' 'mic' 'mare' 'mediu' 'mare' 'mare' 'mare' 'mare'

'mare' 'mediu' 'mare' 'mare' 'mare' 'mare' 'mare' 'mare' 'mare' 'mare'

'mare' 'mare' 'mediu' 'mediu' 'mare' 'mic' 'mic' 'mare' 'mediu' 'mare'

'mediu' 'mic' 'mare' 'mare' 'mare' 'mare' 'mare' 'mic' 'mare' 'mediu'

'mediu' 'mediu']

Test accuracy: 75.00

Train accuracy: 87.30

target = list(casti['Pret'].unique())

feature\_names = list(x.columns)

from sklearn import tree

import graphviz

dot\_data = tree.export\_graphviz(model,

out\_file=None,

feature\_names=feature\_names,

class\_names=target,

filled=True, rounded=True,

special\_characters=True)

graph = graphviz.Source(dot\_data)

graph

from sklearn.tree import export\_text

r = export\_text(model, feature\_names=feature\_names)

print(r)

|--- Frecventa microfon <= 7000.00

| |--- Impedanta <= 18.00

| | |--- class: mare

| |--- Impedanta > 18.00

| | |--- class: mic

|--- Frecventa microfon > 7000.00

| |--- Frecventa casti <= 15000.00

| | |--- class: mic

| |--- Frecventa casti > 15000.00

| | |--- Impedanta <= 30.00

| | | |--- Impedanta <= 18.00

| | | | |--- class: mare

| | | |--- Impedanta > 18.00

| | | | |--- class: mediu

| | |--- Impedanta > 30.00

| | | |--- Brand <= 5.50

| | | | |--- Frecventa casti <= 26000.00

| | | | | |--- Brand <= 3.50

| | | | | | |--- Brand <= 2.50

| | | | | | | |--- Impedanta <= 35.50

| | | | | | | | |--- Impedanta <= 32.50

| | | | | | | | | |--- Frecventa microfon <= 9000.00

| | | | | | | | | | |--- class: mare

| | | | | | | | | |--- Frecventa microfon > 9000.00

| | | | | | | | | | |--- class: mare

| | | | | | | | |--- Impedanta > 32.50

| | | | | | | | | |--- class: mare

| | | | | | | |--- Impedanta > 35.50

| | | | | | | | |--- Frecventa microfon <= 11000.00

| | | | | | | | | |--- Brand <= 1.50

| | | | | | | | | | |--- Impedanta <= 40.50

| | | | | | | | | | | |--- class: mare

| | | | | | | | | | |--- Impedanta > 40.50

| | | | | | | | | | | |--- class: mare

| | | | | | | | | |--- Brand > 1.50

| | | | | | | | | | |--- class: mediu

| | | | | | | | |--- Frecventa microfon > 11000.00

| | | | | | | | | |--- Frecventa microfon <= 16000.00

| | | | | | | | | | |--- class: mare

| | | | | | | | | |--- Frecventa microfon > 16000.00

| | | | | | | | | | |--- class: mare

| | | | | | |--- Brand > 2.50

| | | | | | | |--- class: mare

| | | | | |--- Brand > 3.50

| | | | | | |--- Brand <= 4.50

| | | | | | | |--- Frecventa microfon <= 11500.00

| | | | | | | | |--- Frecventa microfon <= 9000.00

| | | | | | | | | |--- class: mediu

| | | | | | | | |--- Frecventa microfon > 9000.00

| | | | | | | | | |--- class: mediu

| | | | | | | |--- Frecventa microfon > 11500.00

| | | | | | | | |--- class: mic

| | | | | | |--- Brand > 4.50

| | | | | | | |--- Frecventa microfon <= 9100.00

| | | | | | | | |--- class: mare

| | | | | | | |--- Frecventa microfon > 9100.00

| | | | | | | | |--- Frecventa casti <= 20500.00

| | | | | | | | | |--- class: mare

| | | | | | | | |--- Frecventa casti > 20500.00

| | | | | | | | | |--- class: mare

| | | | |--- Frecventa casti > 26000.00

| | | | | |--- Brand <= 2.50

| | | | | | |--- Impedanta <= 34.50

| | | | | | | |--- class: mediu

| | | | | | |--- Impedanta > 34.50

| | | | | | | |--- class: mic

| | | | | |--- Brand > 2.50

| | | | | | |--- Frecventa microfon <= 9000.00

| | | | | | | |--- class: mare

| | | | | | |--- Frecventa microfon > 9000.00

| | | | | | | |--- Frecventa microfon <= 14000.00

| | | | | | | | |--- Frecventa casti <= 33500.00

| | | | | | | | | |--- class: mediu

| | | | | | | | |--- Frecventa casti > 33500.00

| | | | | | | | | |--- Impedanta <= 33.50

| | | | | | | | | | |--- class: mediu

| | | | | | | | | |--- Impedanta > 33.50

| | | | | | | | | | |--- class: mare

| | | | | | | |--- Frecventa microfon > 14000.00

| | | | | | | | |--- class: mare

| | | |--- Brand > 5.50

| | | | |--- class: mare

graph.save('graph1.jpg')

graph1.jpg