به نام خدا

تمرین ۳

علیرضا محمدیان

زهرا حسینی

حسین البکری

سوال ۱-

```
df1 = pd.read_csv(data1 , header=None).replace("5more", "more")
df1.columns = ["buying", "maint", "doors", "persons", "lug_boot", "safety" , "decision"]

#question1
X , y = df1.drop(columns=["decision"],axis = 1) , df1["decision"]
```

سوال ۲ –

```
#question2
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20,random_state=42)
```

سوال ۳ –

نوع همه متغییر ها از نوع object است .

```
print(X_train.dtypes)

for i in range(len(X_train.columns)):
    print(X_train[X_train.columns[i]].unique())
```

```
buying      object
maint       object
doors       object
persons     object
lug_boot    object
safety      object
dtype: object
['vhigh' 'med' 'low' 'high']
['vhigh' 'low' 'med' 'high']
['more' '3' '4' '2']
['more' '4' '2']
['big' 'small' 'med']
['high' 'med' 'low']
```

انکود کردن :

```
encoder = OrdinalEncoder()
encoder.fit(X_train)
X_train = encoder.transform(X_train)
X_test = encoder.transform(X_test)
```

سوال ٤ –

ایجاد درخت تصمیم

```
clf = DecisionTreeClassifier(criterion="entropy")
clf.fit(X_train , y_train)
```

ابتدا accuracy را برای داده های train چاپ میکنیم و سپس classification_report را چاپ میکنیم که شامل روش های ارزیابی دیگر مانند precision و recall است.

دقت کنید ما ماتری confusion را نیز چاپ کردیم

```
pred_train = clf.predict(X_train)
print("accuracy score for train data is : " + str(accuracy_score(y_train , pred_train)))
print("\n")
print(confusion_matrix(y_train , pred_train))
print(classification_report(y_train, pred_train))
```

خروجی

```
accuracy score for train data is : 1.0
[[301   0   0   0]
 [  0  58   0   0]
 [  0   0 975   0]
 [  0   0   0  48]]
              precision    recall  f1-score   support

         acc       1.00      1.00      1.00       301
        good       1.00      1.00      1.00        58
       unacc       1.00      1.00      1.00       975
       vgood       1.00      1.00      1.00        48

    accuracy                           1.00      1382
   macro avg       1.00      1.00      1.00      1382
weighted avg       1.00      1.00      1.00      1382
```

همین کار را برای داده های تست انجام میدهیم

```
#predict the test
pred_test = clf.predict(X_test)
print(confusion_matrix(y_test , pred_test))
print(classification_report(y_test, pred_test))
```

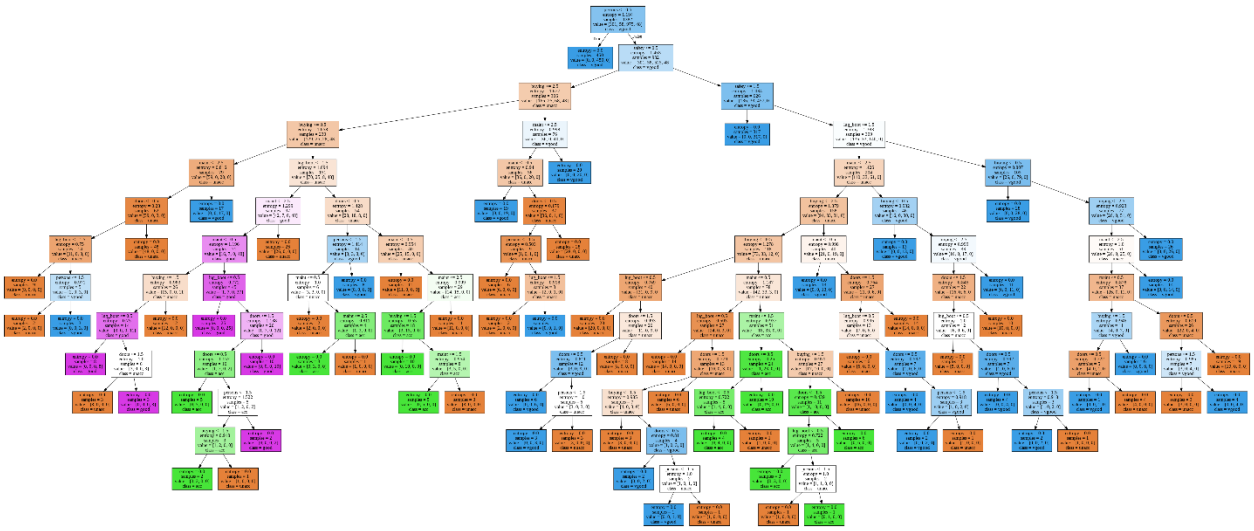
و خروجی

```
[[ 76   6   1   0]
 [  1  10   0   0]
 [  0   0 235   0]
 [  1   0   0  16]]
              precision    recall  f1-score   support

         acc       0.97      0.92      0.94        83
        good       0.62      0.91      0.74        11
       unacc       1.00      1.00      1.00       235
       vgood       1.00      0.94      0.97        17

    accuracy                           0.97       346
   macro avg       0.90      0.94      0.91       346
weighted avg       0.98      0.97      0.98       346
```

در ادامه با استفاده از graphviz درخت را میکشیم :

```
dot_graph = tree.export_graphviz(clf,
                    feature_names = df1.columns[:6],
                    class_names=y.unique(),
                    filled = True)
graph = graphviz.Source(dot_graph , format="png")
graph.render("tree")
```

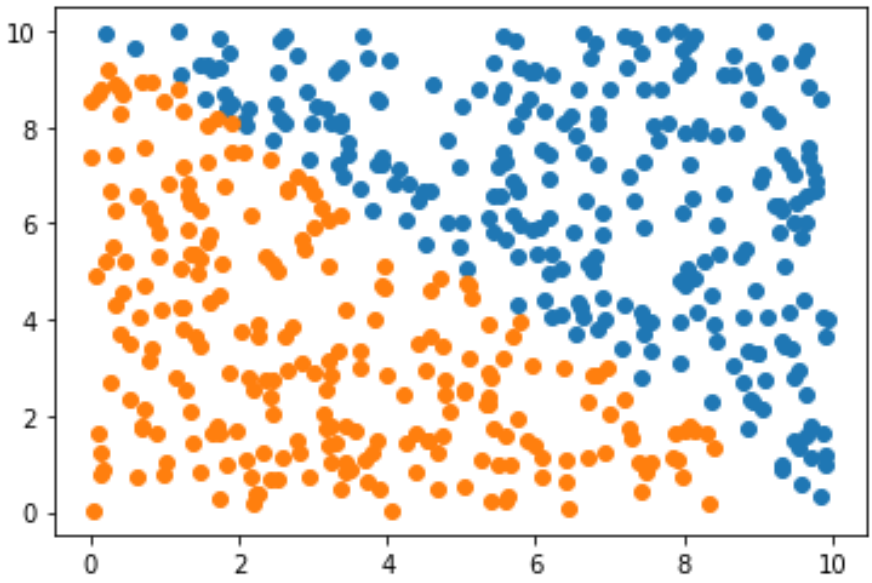
درخت خروجی به صورت زیر است (فایل عکس در فایل زیپ موجود است)



سوال ۵ –

ابتدا داد ها را چاپ میکنیم

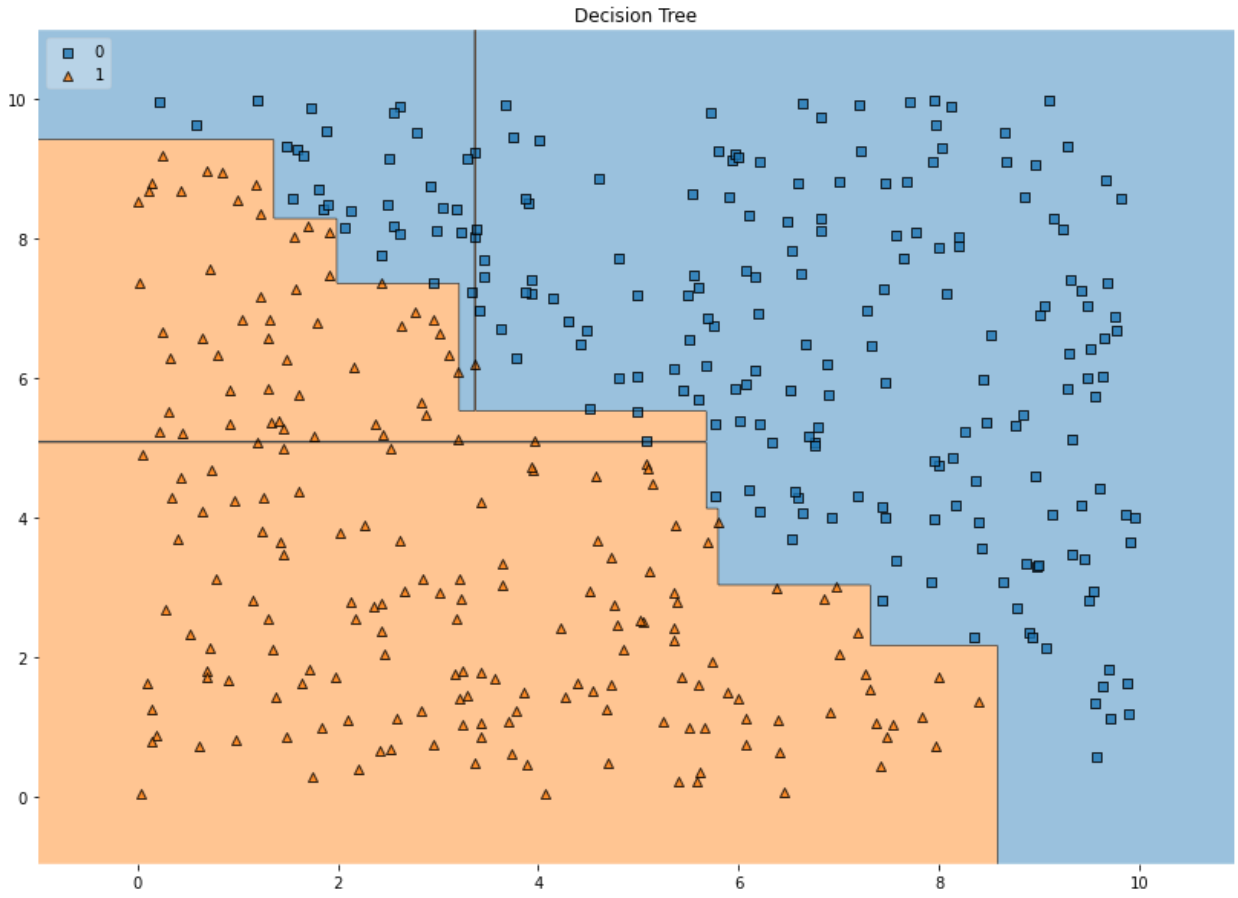
```
zero = df2[df2["label"] == 0]
one = df2[df2["label"] == 1]
plt.scatter(zero["x0"] , zero["x1"] , label = "0")
plt.scatter(one["x0"] , one["x1"] , label = "1")
plt.show()
```



حال خطوط تصمیم درخت را رسم میکنیم توجه کنید ابتدا داده ها را به دو بخش test و train تقسیم کردیم

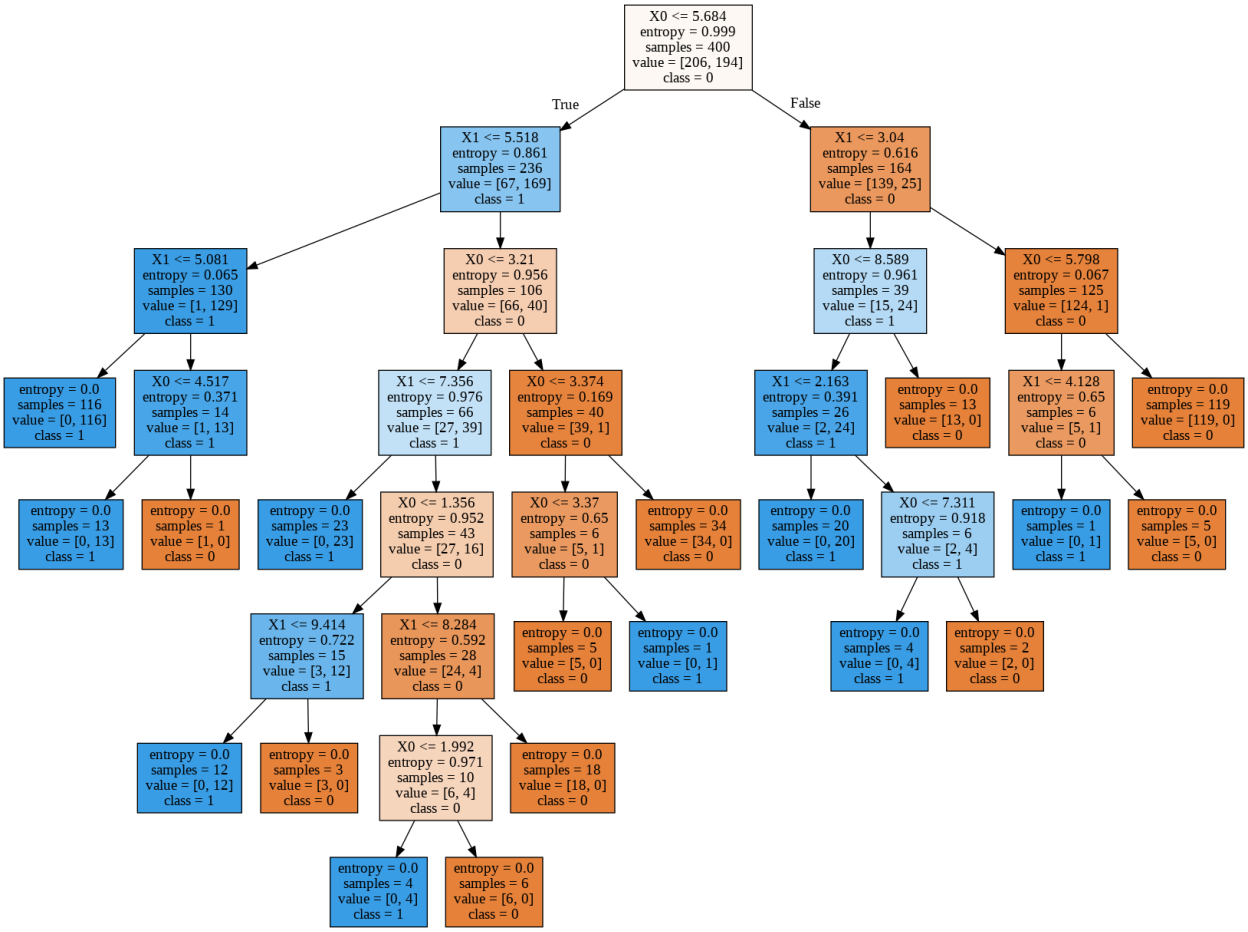
```
X_train,X_test,y_train,y_test=train_test_split(df2.drop(columns=["label"] , axis = 1),df2["label"],test_size=0.20,random_state=42)
clf2 = DecisionTreeClassifier()
fig = plt.figure(figsize=(14,10))
label = 'Decision Tree'
clf.fit(X_train, y_train)
fig = plot_decision_regions(X=np.array(X_train), y=np.array(pd.Categorical(y_train).codes), clf=clf, legend=2)
plt.title(label)
plt.show()
```

خروجی به صورت



حال درخت تصمیم را میکشیم

```
#%%
dot_graph = tree.export_graphviz(clf,
                    feature_names = df1.columns[:6],
                    class_names=y.unique(),
                    filled = True)
graph = graphviz.Source(dot_graph , format="png")
graph.render("tree2")
```



قوانینی که بدست آمده درواقع مسیری است که به یک برگ درخت خاتمه پیدا میکند در واقع هر برگ درخت یک خط تصمیم میباشد که میتوان با آن مرز های دسته بندی را مشخص کرد .به طور مثال این X1<=٥,٠٨١ , X1<=٥,٥١٨ , X٠ <= ٥,٦٨٤

مسیر منجر به شامل شدن ۱۱٦ نمونه از کلاس ۱ میشود حالا به طور معادل برای این حال چون درخت دودویی میباشد . X1= ٥,٠٨١ مسیر یک خط تصمیم وجود دارد خط X1<=٥,٥١٨ , X٠ <= ٥,٦٨٤ بعضی خط ها اورلپ دارند مثلا در مثال قبل خط دقیقا مرز دو کلاس میشود X٠=٤,٥١٧ در این مسیر خط X٠<=٤,٥١٧, X1<=٥,٠٨١ پس دوخط روی هم میفتند یعنی به ازای دو برگ یک خط داریم .به طور کل ۱۹ برگ و ۱۳ خط تصمیم وجود دارد.

: برای بهبود مدل میتوان از راه های زیر استفاده کرد

-k-fold cross validation

ensemble استفاده از درخت تصمیم به صورت

استفاده از شاخص های دیگر مانند جینی