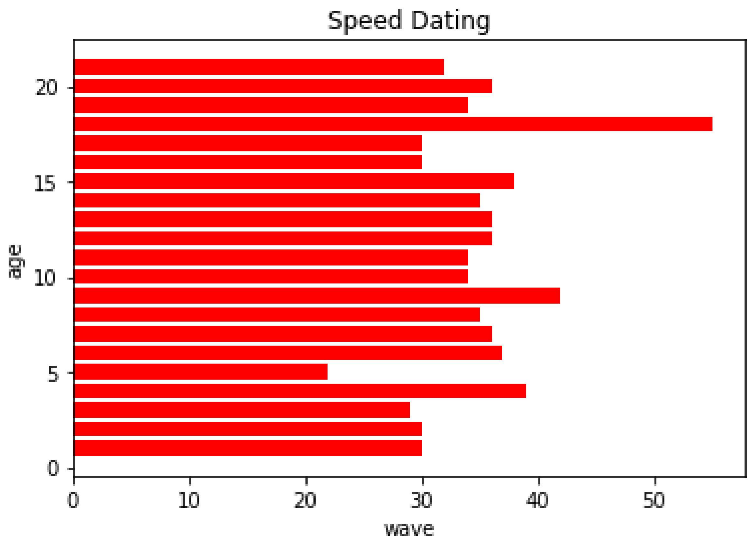
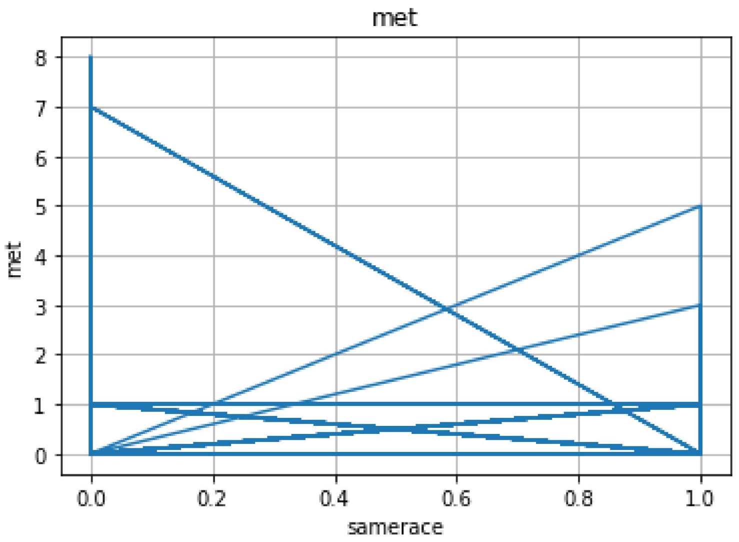
In [86]:
```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.io import arff
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import preprocessing
from sklearn.metrics import accuracy_score, classification_report, confusion_matr
from sklearn.tree import DecisionTreeClassifier
from sklearn import preprocessing
from sklearn import utils
d = arff.loadarff(r'C:\Users\mohamedatham.s\Downloads\speeddating.arff')
df=pd.DataFrame(d[0])
for i,j in df.dtypes.items():
    if j==np.object:
        df[i]=df[i].str.decode('utf-8').fillna(df[i])
df=df.replace('female',0)
df=df.replace('male',1)
le = LabelEncoder()
#encoder
df['race']=le.fit_transform(df['race'])
df['race_o']=le.fit_transform(df['race_o'])
df['has_null']=le.fit_transform(df['has_null'])
df['samerace']=le.fit_transform(df['samerace'])
# normalise
df['race']=df['race']+df['race'].abs().max()
df['race_o']=df['race_o']+df['race_o'].abs().max()
display(df.head(7))
# bar
plt.barh(df['wave'],df['age'],color = "red")
plt.title('Speed Dating')
plt.xlabel('wave')
plt.ylabel('age')
plt.show()
# line
plt.plot(df['samerace'],df['met'])
plt.title('met')
plt.xlabel('samerace')
plt.ylabel('met')
plt.grid()
plt.show()
# pie chart
plt.pie(df['decision'],labels=df['wave'])
plt.title('pie chart')
plt.xlabel('d 1')
plt.ylabel('d 2')
plt.show()
# train_test
x=df.iloc[:,1]
y=df.iloc[:,7]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0
x_test=x_test.values.reshape(-1,1)
x_train=x_train.values.reshape(-1,1)
```

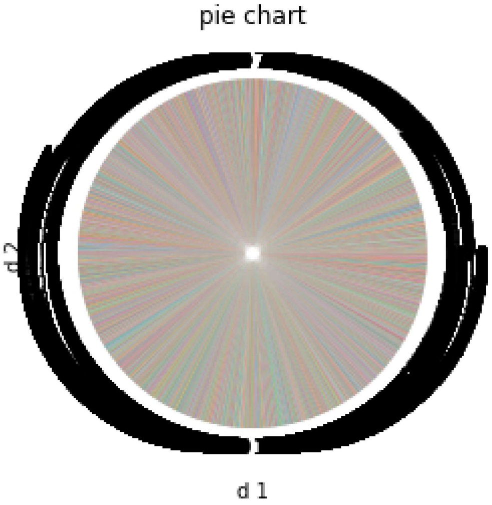
```
l1=preprocessing.LabelEncoder()
ytrain_t=l1.fit_transform(y_train)
print(ytrain_t)
l1=preprocessing.LabelEncoder()
ytest_t=l1.fit_transform(y_test)
print(ytest_t)
cl= DecisionTreeClassifier(criterion='entropy', random_state=0)
cl.fit(x_train, ytrain_t)
y_pred=cl.predict(x_test)
print(accuracy_score(y_true=ytrain_t,y_pred=cl.predict(x_train)))
print(accuracy_score(y_true=ytest_t,y_pred=cl.predict(x_test)))
plt.scatter(x_test,y_test,color='violet')
plt.plot(x_test,y_pred,color='yellow')
plt.title('Decision tree')
plt.show()
```

	has_null	wave	gender	age	age_o	d_age	d_d_age	race	race_o	samerace	...	d_expected_
0	0	1.0	0	21.0	27.0	6.0	[4-6]	6	8	0	...	
1	0	1.0	0	21.0	22.0	1.0	[0-1]	6	8	0	...	
2	1	1.0	0	21.0	22.0	1.0	[0-1]	6	6	1	...	
3	0	1.0	0	21.0	23.0	2.0	[2-3]	6	8	0	...	
4	0	1.0	0	21.0	24.0	3.0	[2-3]	6	9	0	...	
5	0	1.0	0	21.0	25.0	4.0	[4-6]	6	8	0	...	
6	0	1.0	0	21.0	30.0	9.0	[7-37]	6	8	0	...	

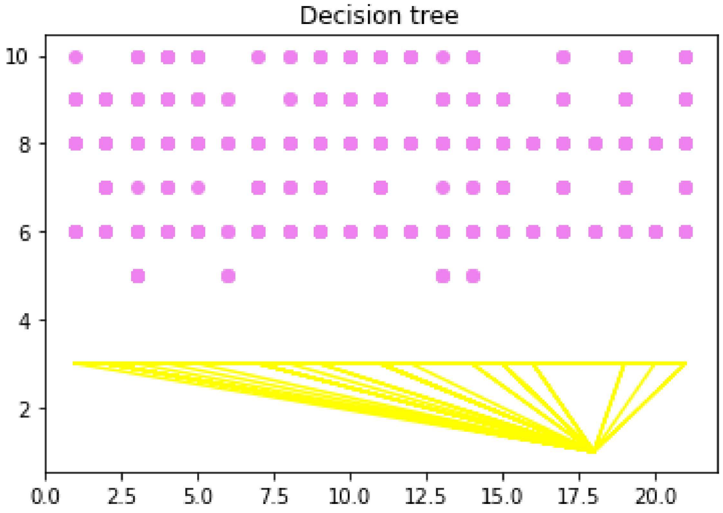
7 rows × 123 columns







```
[3 1 3 ... 1 3 3]
[3 2 1 ... 3 1 5]
0.5677224255928697
0.568019093078759
```



In [ ]:

In [ ]: