Random Forest

import dataset

```
import pandas as pd
df = pd.read_csv("drug200.csv")
df.head()
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY

cleaning

```
# clean the data
```

encoding

```
from sklearn.preprocessing import LabelEncoder

le_sex = LabelEncoder().fit(df['Sex'])
df['Sex'] = le_sex.transform(df['Sex'])

le_BP = LabelEncoder().fit(df['BP'])
df['BP'] = le_BP.transform(df['BP'])

le_Chol = LabelEncoder().fit(df['Cholesterol'])
df['Cholesterol'] = le_Chol.fit_transform(df['Cholesterol'])

df.head()
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	0	0	0	25.355	drugY
1	47	1	1	0	13.093	drugC
2	47	1	1	0	10.114	drugC
3	28	0	2	0	7.798	drugX
4	61	0	1	0	18.043	drugY

define x and y

```
import numpy as np
x = df[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].values
x[0:3]
```

```
array([[23.    ,  0.    ,  0.    ,  0.    , 25.355],
       [47.    ,  1.    ,  1.    ,  0.    , 13.093],
       [47.    ,  1.    ,  1.    ,  0.    , 10.114]])
```

```
y = df["Drug"].values
y[0:3]
```

```
array(['drugY', 'drugC', 'drugC'], dtype=object)
```

spliting

```
### finding best random state

# from sklearn.model_selection import train_test_split
# from sklearn.ensemble import RandomForestClassifier
# from sklearn.metrics import accuracy_score

# import time
# t1 = time.time()
# lst = []
# for i in range(1,10):
#     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=i)
#     rf = RandomForestClassifier(n_estimators=20)
#     rf.fit(x_train,y_train)
#     yhat_test = rf.predict(x_test)
#     acc = accuracy_score(y_test, yhat_test)
#     lst.append(acc)
# t2 = time.time()
# print(f"run time: {round((t2 - t1) / 60 , 0)} min")
# print(f"accuracy_score = {round(max(lst),2)}")
# print(f"random_state = {np.argmax(lst) + 1}")
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=3)
```

scaling

```
# do not need for scaling
```

fit train data

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=20)
rfc.fit(x_train,y_train)
```

```
▾      RandomForestClassifier
RandomForestClassifier(n_estimators=20)
```

predict test data

```
yhat_test = rfc.predict(x_test)
```

```
print (yhat_test [0:5])
print (y_test [0:5])
```

```
['drugY' 'drugX' 'drugX' 'drugX' 'drugX']
['drugY' 'drugX' 'drugX' 'drugX' 'drugX']
```

evaluation

```
from sklearn.metrics import accuracy_score
print("Accuracy_score (train data): ", accuracy_score(y_train, rfc.predict(x_train)))
print("Accuracy_score (test data): ", accuracy_score(y_test, yhat_test))
```

```
Accuracy_score (train data):  1.0
Accuracy_score (test data):  0.98
```

predict new data

```
rfc.predict([[23, 0, 0, 0, 25.355]])
```

```
array(['drugY'], dtype=object)
```

```
rfc.predict([[23, le_sex.transform(['F'])[0], le_BP.transform(['HIGH'])[0], le_Chol.transform(['HIGH'])[0
```

```
array(['drugY'], dtype=object)
```

save the model

```
# import joblib
# joblib.dump(rfc, 'rfc_model.pkl')
```

load the model

```
# import joblib
# rfc = joblib.load('rfc_model.pkl')
```