

```
In [161]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [162]: # Reading the data
df = pd.read_csv('/content/drug200.csv')
# Visualizing the data
df.head()
```

```
Out[162]:
```

	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

```
In [163]: df.isnull().sum()
```

```
Out[163]: Age          0
Sex            0
BP            0
Cholesterol    0
Na_to_K        0
Drug          0
dtype: int64
```

```
In [164]: df['Drug'].unique()
```

```
Out[164]: array(['DrugY', 'drugC', 'drugX', 'drugA', 'drugB'], dtype=object)
```

```
In [165]: #Finding the count of unique observations
df['Drug'].value_counts()
```

```
Out[165]: DrugY      91
drugX      54
drugA      23
drugC      16
drugB      16
Name: Drug, dtype: int64
```

```
In [166]: from sklearn import preprocessing

label_encoder = preprocessing.LabelEncoder()

# Encode labels for sex - male, female
df['Sex'] = label_encoder.fit_transform(df['Sex'])
```

```
In [167]: label_encoder = preprocessing.LabelEncoder()

df['BP'] = label_encoder.fit_transform(df['BP'])
```

```
In [168]: label_encoder = preprocessing.LabelEncoder()

df['Cholesterol'] = label_encoder.fit_transform(df['Cholesterol'])
```

```
In [169]: df
```

```
Out[169]:
```

	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
...	...	...	...	...	...	...
195	56	0	1	0	11.567	drugC
196	16	1	1	0	12.006	drugC
197	52	1	2	0	9.894	drugX
198	23	1	2	1	14.020	drugX
199	40	0	1	1	11.349	drugX

200 rows × 6 columns

X

[illegible]

```
y
# 5 values...
```

$$[1 \quad 0 \quad 0 \quad 0 \quad 0]$$

```
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,rand
```

```
In [174]: xtrain.shape, xtest.shape, ytrain.shape, ytest.shape
```

```
Out[174]: ((160, 5), (40, 5), (160, 5), (40, 5))
```

```
In [175]: from tensorflow.keras.models import Sequential
          from tensorflow.keras.layers import Dense
```

```
In [189]: model = Sequential()
          model.add(Dense(8, input_dim=5, activation='relu'))
          model.add(Dense(32, activation='relu'))
          model.add(Dense(16, activation='relu'))
          model.add(Dense(26, activation='relu'))
          model.add(Dense(5, activation='softmax'))
```

```
In [190]: model.compile(optimizer='adam', loss='categorical_crossentropy', metr
```

```
In [191]: model.summary()
```

Model: "sequential\_8"

Layer (type)	Output Shape	Param #
dense_39 (Dense)	(None, 8)	48
dense_40 (Dense)	(None, 32)	288
dense_41 (Dense)	(None, 16)	528
dense_42 (Dense)	(None, 26)	442
dense_43 (Dense)	(None, 5)	135
Total params: 1,441		
Trainable params: 1,441		
Non-trainable params: 0		

In [193]: `model.fit(xtrain,ytrain,epochs=200,batch_size=15,validation_data=(x`

```
Epoch 1/200
11/11 [=====] - 0s 10ms/step - loss: 0.23
10 - accuracy: 0.9187 - val_loss: 0.3906 - val_accuracy: 0.8500
Epoch 2/200
11/11 [=====] - 0s 8ms/step - loss: 0.195
2 - accuracy: 0.9438 - val_loss: 0.3833 - val_accuracy: 0.8500
Epoch 3/200
11/11 [=====] - 0s 6ms/step - loss: 0.190
2 - accuracy: 0.9312 - val_loss: 0.4534 - val_accuracy: 0.8000
Epoch 4/200
11/11 [=====] - 0s 6ms/step - loss: 0.196
9 - accuracy: 0.9312 - val_loss: 0.4746 - val_accuracy: 0.8250
Epoch 5/200
11/11 [=====] - 0s 6ms/step - loss: 0.185
1 - accuracy: 0.9500 - val_loss: 0.4110 - val_accuracy: 0.8250
Epoch 6/200
11/11 [=====] - 0s 7ms/step - loss: 0.188
2 - accuracy: 0.9438 - val_loss: 0.4034 - val_accuracy: 0.8500
Epoch 7/200
11/11 [=====] - 0s 6ms/step - loss: 0.184
```

In [196]: `model.predict([[23,1.0,1.0,0.0,23.55]])`

```
1/1 [=====] - 0s 42ms/step
```

Out[196]: `array([[9.9272656e-01, 6.7993609e-13, 3.0613254e-08, 7.2733914e-03`  
`, 2.5104456e-09]], dtype=float32)`

In [197]: `model.predict([[50,0,2,1,7.285]])`

```
1/1 [=====] - 0s 44ms/step
```

Out[197]: `array([[1.9470191e-07, 3.2284796e-16, 1.5296705e-11, 3.2615176e-07`  
`, 9.9999952e-01]], dtype=float32)`