

AIM

Programs on feedforward network to classify any standard dataset available in the public domain.

Dataset used: HR_comma_sep.csv

Programming code:

```
import numpy as np
import pandas as pd

# Load data
data=pd.read_csv('HR_comma_sep.csv')
data.head()
```

OUTPUT:

	satisfaction_level	last_evaluation	number_project	average_monthly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

Programming code:

```
from sklearn import preprocessing
# Creating labelEncoder
le = preprocessing.LabelEncoder()
# Converting string labels into numbers.
data['salary']=le.fit_transform(data['salary'])
data['sales']=le.fit_transform(data['sales'])
```

```
X=data[['satisfaction_level', 'last_evaluation', 'number_project', 'average_monthly_hours',  
'time_spend_company', 'Work_accident', 'promotion_last_5years', 'sales', 'salary']]
```

```
y=data['left']
```

```
# Import train_test_split function
```

```
from sklearn.model_selection import train_test_split
```

```
# Split dataset into training set and test set
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
# 70% training and 30% test
```

```
from sklearn.neural_network import MLPClassifier
```

```
# Create model object
```

```
clf = MLPClassifier(hidden_layer_sizes=(6,5),
```

```
                    random_state=5,
```

```
                    verbose=False,
```

```
                    learning_rate_init=0.01)
```

```
# Fit data onto the model
```

```
clf.fit(X_train,y_train)
```

OUTPUT:

```
MLPClassifier(hidden_layer_sizes=(6, 5), learning_rate_init=0.01,  
              random_state=5)
```

Programming code:

```
ypred=clf.predict(X_test)
```

```
# Import accuracy score
```

```
from sklearn.metrics import accuracy_score
```

```
# Calculate accuracy
```

```
accuracy_score(y_test,ypred)
```

```
0.9386666666666666
```
