

سید محمد طاهل طباطبایی - گزارش

## بخش اول

برای بخش اول، ابتدا داده‌های ترین را با کمک لایبرری پانداس میخوانیم.

```
import pandas as pd
data = pd.read_csv('train.csv')
data.head()
```

[100] ✓ 0.2s

با توجه به شکل دیتا، داده‌های ترین را نویزگیری می‌کنیم و ستون‌های بلااستفاده را حذف و لیبل‌های رشته‌ای را به عدد تغییر می‌دهیم. دو عکس از شکل داده قبل و بعد از پردازش داده‌ها در کد قابل مشاهده است.

```
data = data.drop(['Unnamed: 0', 'id'], axis=1).dropna()

from sklearn.preprocessing import LabelEncoder
for column in ['Gender', 'Customer Type', 'Type of Travel', 'Class', 'satisfaction']:
    encoder = LabelEncoder()
    data[column] = encoder.fit_transform(data[column])
```

در انتها نیز داده‌ها و لیبل‌ها را در دو متغیر X, Y ذخیره می‌کنیم.

```
x = data.drop('satisfaction', axis=1)
y = data['satisfaction']
```

حال با استفاده از لایبرری sklearn ، یک پرسپترون ساده می‌سازیم و با داده‌های ترین عمل فیت را انجام می‌دهیم.

```
from sklearn.linear_model import Perceptron
trainPer = Perceptron(random_state=0)
trainPer.fit(x.values, y.values)
```

[102] ✓ 0.7s

... Perceptron()

پس از آن داده‌های تست را نیز مانند بالا می‌خوانیم.

```
dataTest = pd.read_csv('test.csv')
dataTest.head()

dataTest = dataTest.drop(['Unnamed: 0', 'id'], axis=1).dropna()

from sklearn.preprocessing import LabelEncoder
for column in ['Gender', 'Customer Type', 'Type of Travel', 'Class', 'satisfaction']:
    encoder = LabelEncoder()
    dataTest[column] = encoder.fit_transform(dataTest[column])

x2 = dataTest.drop('satisfaction', axis=1)
y2 = dataTest['satisfaction']
```

[103] ✓ 0.1s

حالا از پرسپترون مرحله قبل استفاده می‌کنیم و دقت آن را روی داده‌های تست می‌سنجیم.

```
from sklearn.metrics import accuracy_score
accuracy_score(y2.values, trainPer.predict(x2.values))
```

[104] ✓ 0.3s

... 0.8280616382806164

## بخش دوم

در این بخش، از الگوریتم RBF برای انتقال داده‌ها به بعد جدید و از تابع SGD برای دسته‌بندی داده‌ها با ابعاد جدید استفاده می‌کنیم.

```
from sklearn.kernel_approximation import RBFSampler
from sklearn.linear_model import SGDClassifier

rbf_feature = RBFSampler(gamma=0.0001, random_state=0)
X_features = rbf_feature.fit_transform(dataTest)

clf = SGDClassifier(max_iter=2000)
clf.fit(X_features, y2.values)
clf.score(X_features, y2.values)
```

[131] ✓ 0.2s

... 0.6602942880315144