



دوره جامع پایتون:
بخش یادگیری ماشین
جلسه بیستم و یکم

دکتر ذبیح اله ذبیحی

مثال

- پیش بینی نمره دانشجویان بر اساس مدت ساعت مطالعه

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
dataset = pd.read_csv('student_scores.csv')
Print(dataset.shape)
Print(dataset)
Print(dataset.describe())
dataset.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
print(model.intercept_)
print(model.coef_)
y_pred = model.predict(X_test)
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(df)
```

```
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test,
y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test,
y_pred))

print('Root Mean Squared Error:',
np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

$$\text{Mean Absolute Error (MAE)} = \frac{1}{n} \sum_{i=1}^n |actual - predicted|$$

$$\text{Mean Squared Error (MSE)} = \frac{1}{n} \sum_{i=1}^n |actual - predicted|^2$$

$$\text{Root Mean Squared Error (RMSE)} = \sqrt{\frac{1}{n} \sum_{i=1}^n |actual - predicted|^2}$$

مثال: نحوه تحلیل خروجی

Mean: 51.48

Mean Absolute Error: 4.183859899

Mean Squared Error: 21.5987693072

Root Mean Squared Error: 4.6474476121

You can see that the value of root mean squared error is 4.64, which is less than 10% of the mean value of the percentages of all the students i.e. 51.48. This means that our algorithm did a decent job.

مثال: نحوه تحلیل خروجی

Mean: 22.41

('Mean Absolute Error:', 3.19932917837853)

('Mean Squared Error:', 17.631568097568447)

('Root Mean Squared Error:', 4.198996082109204)

You can see that the value of root mean squared error is 4.19, which is more than 10% of the mean value of the percentages of all the temperature i.e. 22.41. This means that our algorithm was not very accurate but can still make reasonably good predictions.

مثال

- با توجه به دیتاست، چه سائز لباسی مناسب فردی با وزن 69kg و قد 169cm است.

Height	Weight	T-Shirt Size
150	51	M
158	51	M
158	53	M
158	55	M
159	55	M
159	56	M
160	57	M
160	58	M
160	58	M
162	52	L
163	53	L
165	53	L
167	55	L
168	62	L
168	65	L
169	67	L
169	68	L
170	68	L
170	69	L

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
dataset = pd.read_excel('T_shirt.xlsx')
print(dataset.shape)
print(dataset)
print(dataset.describe())
dataset.plot(x='Height', y='Weight', style='o')
plt.title('Height vs Weight')
plt.xlabel('Height')
plt.ylabel('Weight')
```

```
X = dataset.iloc[:, :-1].values
```

```
y = dataset.iloc[:, 2].values
```

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

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
model = KNeighborsClassifier(n_neighbors=1)
```

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

```
y_pred=model.predict(X_test)
```

```
print(X_test)
```

```
print(y_test)
```

```
print(y_pred)
```

```
import numpy as np
error_rate = []
# Might take some time
for i in range(1,15):

    model = KNeighborsClassifier(n_neighbors=i)
    model.fit(X_train,y_train)
    pred_i = model.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
plt.plot(range(1,15),error_rate,color='blue', linestyle='dashed',
marker='o',
        markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
plt.show()
```

```
y_new=model.predict([[69,169]])  
print(y_new)
```


مثال: پیش بینی قیمت خانه

قیمت f	منطقه e	سال ساخت d	طبقه c	تعداد اتاق b	متراژ a	
۶۲۰	۱	۱۰	۴	۲	۶۰	۱
۹۲۰	۱	۵	۲	۲	۸۰	۲
۳۶۰	۱	۱۵	۱	۱	۴۰	۳
۱۱۲۵	۲	۵	۲	۲	۷۵	۴
۱۱۷۰	۲	۱۰	۴	۳	۹۰	۵
۵۶۰	۲	۲	۱	۱	۳۵	۶
۷۰۰	۲	۱۰	۳	۱	۵۰	۷
۱۶۰۰	۳	۵	۵	۲	۸۰	۸
۱۸۰۰	۳	۱۰	۴	۳	۱۰۰	۹
۱۰۰۰	۳	۲	۱	۱	۵۰	۱۰

```
import pandas as pd
data = {"a":[60,80,40,75,90,35,50,80,100,50], "b":[2,2,1,2,3,1,1,2,3,1], "c":[4,2,1,2,4,1,3,5,4,1],
        "d":[10,5,15,5,10,2,10,5,10,2], "e":[1,1,1,2,2,2,2,3,3,3], "f":[620,920,360,1125,1170,560,700,1600,1800,1000]}
df= pd.DataFrame(data,columns= ['a', 'b','c','d','e','f'])
X = df[['a', 'b','c','d','e']]
y = df['f']
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
X_train,X_test,y_train,y_test= train_test_split(X,df['f'],test_size=0.10)
model = KNeighborsClassifier(n_neighbors=1)
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
print(X_test)
print(y_pred)
```

مثال: اگر یک محصول با وزن ۱۲۵ ، اندازه ۴۵ و رنگ آبی دیده شود مربوط به کدام دسته A یا B خواهد بود.

برحسب	رنگ	اندازه (سانتی متر)	وزن (کیلوگرم)	نمونه
A	1	50	120	1
B	2	20	60	2
A	1	65	145	3
A	3	45	130	4
B	2	15	50	5

روش یادگیری با نظارت، کلاس بندی، تکنیک KNN

```
import numpy as np
dataset = np.array([[120, 50, 1, 0],[60, 20, 2, 1],
                    [145, 65, 1, 0],[130, 45, 3, 0],
                    [50, 15, 2, 1]])
features = dataset[:, :3]
label = dataset[:, 3]

from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=3)
model.fit(features, label)
new_sample = [[125, 45, 1]]

resulted_class = model.predict(new_sample)
print(resulted_class)
```