بسمه تعالی

ارائه دهنده : زهرا منصوری

عنوان : کارگاه یادگیری ماشین

فهرست

تحلیل مجموعه داده

مجموعه داده انتخابی

فایل داده شبکه انتخاب شده با نام ’dataR2.csv‘ در پوشه پروژه وجود دارد. این مجموعه داده شامل اطلاعاتی نظیر سن، BMI، Glucose، Insulin، HOMA، Leptin، Adiponectin، Resistin، MCP.1 است که از گروهی از افراد به دست آمده و در نهایت به دو کلاس مبتلا به سرطان و یا سالم دسته بندی شده اند.

```
In [1]: import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        import numpy as np
```

```
In [2]: df = pd.read_csv('dataR2.csv')
        df.round(2)
```

Out[2]:

	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1	Classification
0	48	23.50	70	2.71	0.47	8.81	9.70	8.00	417.11	1
1	83	20.69	92	3.12	0.71	8.84	5.43	4.06	468.79	1
2	82	23.12	91	4.50	1.01	17.94	22.43	9.28	554.70	1
3	68	21.37	77	3.23	0.61	9.88	7.17	12.77	928.22	1
4	86	21.11	92	3.55	0.81	6.70	4.82	10.58	773.92	1
...	...	...	...	...	...	...	...	...	...	...
111	45	26.85	92	3.33	0.76	54.68	12.10	10.96	268.23	2
112	62	26.84	100	4.53	1.12	12.45	21.42	7.32	330.16	2
113	65	32.05	97	5.73	1.37	61.48	22.54	10.33	314.05	2
114	72	25.59	82	2.82	0.57	24.96	33.75	3.27	392.46	2
115	86	27.18	138	19.91	6.78	90.28	14.11	4.35	90.09	2

116 rows × 10 columns

ابعاد داده ها

در این مجموعه داده، ۱۱۶ نمونه با ۹ ویژگی داریم.

```
In [3]: data = df[["Age","BMI","Glucose","Insulin","HOMA","Leptin","Adiponectin","Resistin","MCP.1"]]
        print(data.shape)
```

```
(116, 9)
```

توزیع کلاس ها

بر اساس ستون Classification، نمونه ها را گروه بندی می کنیم. طبق خروجی، ۵۲ نمونه در گروه ۱ و ۶۴ نمونه در گروه ۲ قرار می گیرند.

```
In [4]: count_class = df.groupby("Classification").size()
        print(count_class)
```

```
Classification
1    52
2    64
dtype: int64
```

نوع ویژگی ها

```
In [5]: print(data.dtypes)
```

```
Age              int64
BMI            float64
Glucose          int64
Insulin        float64
HOMA           float64
Leptin         float64
Adiponectin    float64
Resistin       float64
MCP.1          float64
dtype: object
```

همبستگی ویژگی ها

همانطور که مشاهده می شود مقدار همبستگی Insulin و HOMA، ۰,۹۳ می باشد و بسیار نزدیک به ۱ است که نشان دهنده رابطه مستقیم این دو ویژگی به هم دارد.

بسیاری از ویژگی ها نسبت به هم، همبستگی ندارند؛ در نتیجه مقدار correlation آنها ۰ یا نزدیک به آن است.

منفی ترین مقدار correlation نیز مربوط به دو ویژگی Adiponectin و BMI است که مقدار آن ۰,۳- می باشد و نشانگر همبستگی در جهت مخالف هم دیگر است.

```
In [6]: correlations = data.corr(method='pearson')
        correlations
```

Out[6]:

	Age	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1
Age	1.000000	0.008530	0.230106	0.032495	0.127033	0.102626	-0.219813	0.002742	0.013462
BMI	0.008530	1.000000	0.138845	0.145295	0.114480	0.569593	-0.302735	0.195350	0.224038
Glucose	0.230106	0.138845	1.000000	0.504653	0.696212	0.305080	-0.122121	0.291327	0.264879
Insulin	0.032495	0.145295	0.504653	1.000000	0.932198	0.301462	-0.031296	0.146731	0.174356
HOMA	0.127033	0.114480	0.696212	0.932198	1.000000	0.327210	-0.056337	0.231101	0.259529
Leptin	0.102626	0.569593	0.305080	0.301462	0.327210	1.000000	-0.095389	0.256234	0.014009
Adiponectin	-0.219813	-0.302735	-0.122121	-0.031296	-0.056337	-0.095389	1.000000	-0.252363	-0.200694
Resistin	0.002742	0.195350	0.291327	0.146731	0.231101	0.256234	-0.252363	1.000000	0.366474
MCP.1	0.013462	0.224038	0.264879	0.174356	0.259529	0.014009	-0.200694	0.366474	1.000000

نمودار scatter بین ویژگی ها

نمودارهای زیر، پراکندگی نمونه ها را بر اساس ویژگی های موجود در dataset مشخص می کند و نشان می دهد کدام دو ویژگی توانایی ایجاد تمایز بیش تری بین کلاس های ۱ و ۲ دارند.

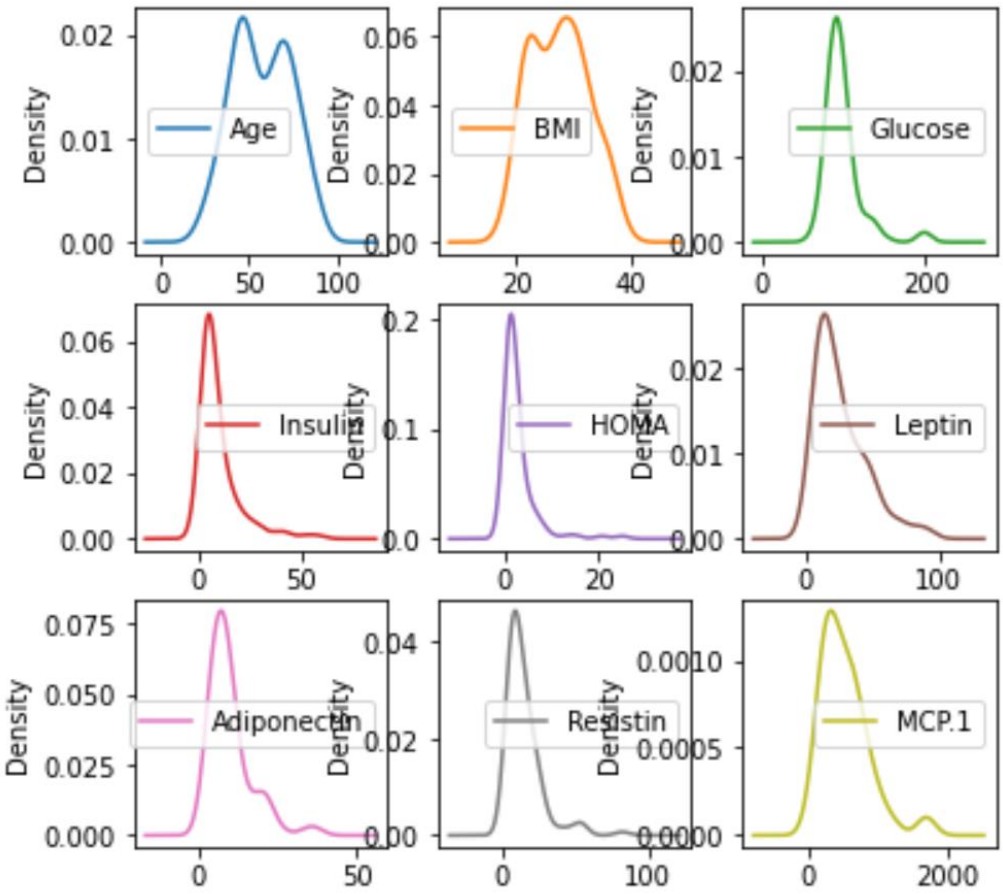
همانطور که در بخش قبل گفته شد، منفی ترین مقدار correlation مربوط به دو ویژگی Adiponectin و BMI است؛ در نتیجه این دو ویژگی بیشترین تمایز را می توانند ایجاد کنند.

```
In [7]: sns.pairplot(df, hue="Classification")
```

```
Out[7]: <seaborn.axisgrid.PairGrid at 0x1b208640>
```

نمودار density ویژگی ها

```
In [8]: data.plot(kind='density', subplots=True, layout=(3, 3), figsize=(6, 6), sharex=False)
```



مسئله classification

محاسبه confusion matrix و accuracy, precision, recall با همه ویژگی ها

ابتدا ۷۰ درصد از نمونه ها را برای آموزش به learner خود داده و ۳۰ درصد از آنها را برای تست learner نگه می داریم.

در confusion matrix به دست آمده، TN برابر با ۷ و FP برابر با ۵ است. مقادیر FN و TP نیز به ترتیب برابر با ۳ و ۲۰ می باشند و در کل ۲۷ (TN + TP) نمونه از ۳۵ نمونه متعلق به داده تست، درست دسته بندی شده اند و باعث ایجاد accuracy حدود ۷۷ درصد شده است؛ که نشان دهنده دقت ۷۷ درصدی learner ما می باشد.

Precision کلاس های یک و دو نیز به ترتیب ۷۰ و ۸۰ درصد محاسبه شده است؛ که نشان دهنده تعداد بالای TN و TP در مقابل مقدار پیشبینی شده از T یعنی (TP + FP) است.

برای محاسبه recall نیز مقدار TP را به مقدار واقعی از کلاس T یعنی (TP + FN) به دست می آوریم؛ که مقدار آن برای کلاس های ۱ و ۲ به ترتیب ۵۸ و ۷۸ درصد است.

In [9]:
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier

x = df.iloc[:, :9].values
y = df.iloc[:, 9].values
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 1)
```

In [10]:
```
clf = DecisionTreeClassifier()
clf = clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
result = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(result)
result1 = classification_report(y_test, y_pred)
print("Classification Report:",)
print (result1)
result2 = accuracy_score(y_test,y_pred)
print("Accuracy:",result2)
```

```
Confusion Matrix:
[[ 7  5]
 [ 3 20]]
Classification Report:
              precision    recall  f1-score   support

           1       0.70      0.58      0.64        12
           2       0.80      0.87      0.83        23

    accuracy                           0.77        35
   macro avg       0.75      0.73      0.73        35
weighted avg       0.77      0.77      0.77        35

Accuracy: 0.7714285714285715
```

محاسبه confusion matrix و accuracy, precision, recall با بخشی از ویژگی ها

در این قسمت تنها از ۶ ویژگی dataset خود استفاده کردیم و ۳ ویژگی را دور ریختیم اما با این حال accuracy حدود ۷۱ درصد به دست آمد که نزدیک به accuracy اصلی می باشد و این موضوع نشان دهنده آن است که ۳ ویژگی حذف شده (Adiponectin, Resistin, MCP.1) نقش اصلی در کلاس بندی نمونه های ما نداشته اند.

In [11]:
```
x = df.iloc[:, :6].values
y = df.iloc[:, 9].values
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 1)
```

In [12]:
```
clf = DecisionTreeClassifier()
clf = clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
result = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(result)
result1 = classification_report(y_test, y_pred)
print("Classification Report:",)
print (result1)
result2 = accuracy_score(y_test,y_pred)
print("Accuracy:",result2)
```

```
Confusion Matrix:
[[ 6  6]
 [ 4 19]]
Classification Report:
              precision    recall  f1-score   support

           1       0.60      0.50      0.55        12
           2       0.76      0.83      0.79        23

    accuracy                           0.71        35
   macro avg       0.68      0.66      0.67        35
weighted avg       0.71      0.71      0.71        35

Accuracy: 0.7142857142857143
```

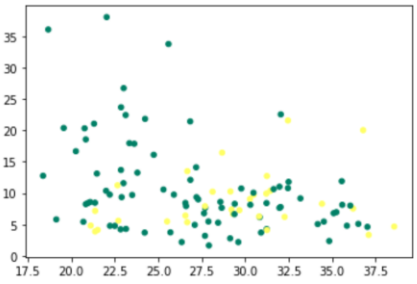
مسئله clustering

در این مسئله می دانیم که دو کلاس داریم اما نمی دانیم هر نمونه مربوط به کدام کلاس است و در این مسئله قرار است نمونه ها را در دو خوشه دسته بندی کنیم.

همانطور که در قسمت محاسبه correlation گفته شد؛ منفی ترین مقدار مربوط به دو ویژگی Adiponectin و BMI است؛ در نتیجه این دو ویژگی بیشترین تمایز را می توانند ایجاد کنند. به همین دلیل نمودار پراکندگی بر اساس این دو ویژگی (۱ و ۶) کشیده شده اند و چون مقدار correlation آنها تقریبا برابر با ۰,۳- است و از ۱- دور می باشد؛ باعث ایجاد تمایز زیادی بین دسته ها نشده است.

In [13]:
```
from sklearn.cluster import KMeans

x = df.iloc[:, :9].values
kmeans = KMeans(n_clusters = 2)
kmeans.fit(x)
y_kmeans = kmeans.predict(x)
```

In [14]:
```
plt.scatter(x[:, 1], x[:, 6], c = y_kmeans, s = 20, cmap = 'summer')
centers = kmeans.cluster_centers_
plt.show()
```



مسئله regression

گزارش RMSE به ازای مدل رگراسیون خطی

برای مسئله رگراسیون، ویژگی HOMA را به عنوان y و بقیه ویژگی ها را به عنوان x در نظر گرفتیم و ۳۰ درصد از نمونه های خود را برای تست و ۷۰ درصد مانده را برای آموزش در نظر گرفتیم.

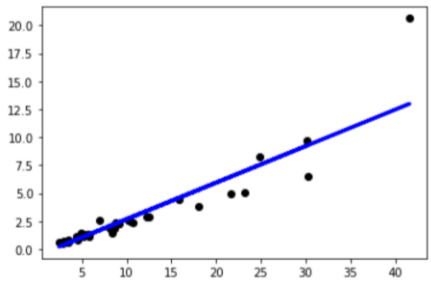
In [25]:
```
x = df.iloc[:, 3].values.reshape(-1, 1)
y = df.iloc[:, 4].values.reshape(-1, 1)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 1)

from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression

regressor = LinearRegression()
regressor.fit(x_train, y_train)
y_pred = regressor.predict(x_test)
```

In [27]:
```
import math
# RMSE for LinearRegression
s = 0
for i in range(len(y_pred)):
    s += ((y_test[i] - y_pred[i]) ** 2)

print(math.sqrt(s/len(y_pred)))
```

```
1.499550096397054
```

نمودار رگراسیون خطی

```
In [28]: plt.scatter(x_test,  y_test,  color='black')
         plt.plot(x_test, y_pred, color='blue', linewidth=3)


         plt.show()
```



گزارش RMSE به ازای مدل رگراسیون چند جمله ای

مقدار RMSE در رگراسیون چند جمله ای با درجه ۳ کمتر از RMSE در مدل خطی شده است و این موضوع نشان می دهد که مدل چند جمله ای با دقت بیش تری عمل می کند.

```
[29]: from sklearn.preprocessing import PolynomialFeatures
      x = df.iloc[:, 3].values.reshape(-1, 1)
      y = df.iloc[:, 4].values.reshape(-1, 1)

      x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 1)

      poly = PolynomialFeatures(degree = 3)
      X_poly = poly.fit_transform(x)

      poly.fit(X_poly, y)
      lin2 = LinearRegression()
      lin2.fit(X_poly, y)
      y_pred = lin2.predict(poly.fit_transform(x_test))

      # RMSE for PolynomialRegression
      s = 0
      for i in range(len(y_pred)):
          s += ((y_test[i] - y_pred[i]) ** 2)

      print(math.sqrt(s/len(y_pred)))
```

```
1.2529003337764746
```

نمودار رگراسیون چند جمله ای

```
In [30]: plt.scatter(x_test, y_test, color = 'blue')
         plt.scatter(x_test, y_pred, color = 'red')
```

```
Out[30]: <matplotlib.collections.PathCollection at 0x22638f10>
```