آموزش رگرسیون خطی (Linear regression)

مقدمه ای بر رگرسیون خطی (Linear regression) :

رگرسیون خطی به عنوان یک مدل آماری تعریف می شود که رابطه خطی بین یک متغیر وابسته و مجموعه متغیر های مستقل داده شده را تحلیل و بررسی می کند. رابطه خطی بین متغیر ها به این معنا است که زمانی که مقدار یک متغیر مستقل یا بیشتر تغییر کند (افزایش یا کاهش) ، مقدار متغیر وابسته نیز متعاقبا تغییر خواهد کرد (افزایش یا کاهش).

عبارت ریاضی زیر رابطه مذکور را در قالب ریاضی ارائه می کند.

$$Y=mX+b$$

در اینجا، Y متغیر وابسته ای است که سعی در پیش بینی آن داریم.

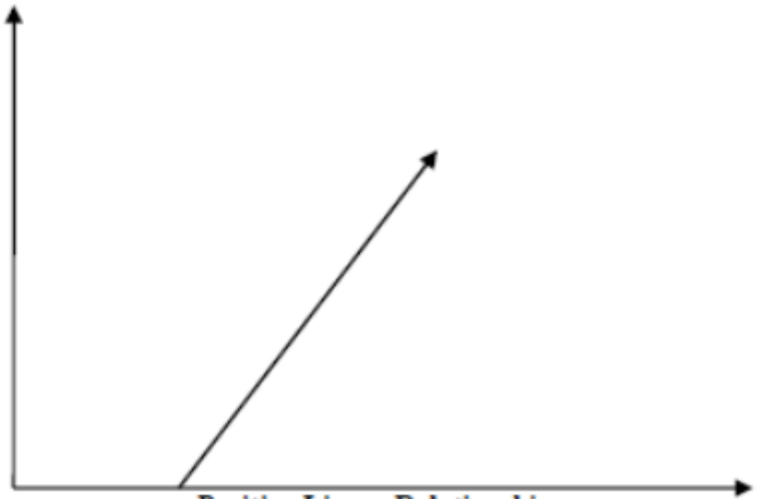
X متغیر مستقلی است که با استفاده از آن پیش بینی را انجام می دهیم.

m شیب خط رگرسیون است که تاثیر X روی Y را نشان می دهد.

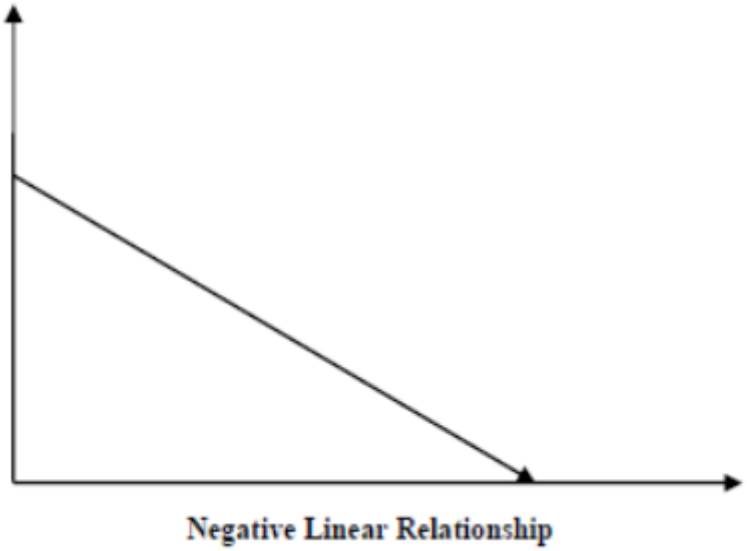
b یک ثابت است که به عنوان YY-intercept شناخته می شود. اگر X=0 ، Y با bb برابر خواهد بود.

علاوه بر این، همانطور که در ادامه توضیح داده می شود، این رابطه خطی می تواند ذاتا مثبت یا منفی باشد.

رابطه خطی مثبت: یک رابطه خطی زمانی مثبت در نظر گرفته می شود که هر دو متغیر وابسته و مستقل افزایش یابند. با کمک نمودار زیر میتوان این موضوع را درک کرد.



رابطه خطی منفی: یک رابطه خطی زمانی منفی در نظر گرفته می شود که متغیر مستقل افزایش و متغیر وابسته کاهش یابند. با کمک نمودار زیر میتوان این موضوع را درک کرد.



انواع رگرسیون خطی: رگرسیون خطی دارای دو نوع رگرسیون خطی ساده (Simple Linear Regression) و رگرسیون خطی چندتایی(Multiple Linear Regression) است.

رگرسیون خطی ساده (SLR): ابتدایی ترین نوع رگرسیون خطی است که تنها با استفاده از یک ویژگی، پاسخ را پیش بینی می کند. فرضیه موجود در SLR این است که دو متغیر با هم رابطه خطی دارند.

پیاده سازی در پایتون: به دو روش می توان SLR را در پایتون پیاده سازی کرد. یک روش این است که مجموعه داده خود را فراهم کنید، و دیگری اینکه از مجموعه داده کتابخانه scikit-learn پایتون استفاده کنید.

مثال ۱ : در مثال زیر از پیاده سازی پایتون، از مجموعه داده خود استفاده می کنیم. ابتدا، به صورت زیر با وارد کردن بسته های ضروری شروع می کنیم.

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

سپس، یک تابع که مقادیر مهم را برای SLR محاسبه می کند، تعریف کنید.

```
def coef_estimation(x, y):
```

اسکریپت زیر تعداد مشاهدات n را ارائه می کند.

```
n = np.size(x)
```

میانگین بردار های xو y به صورت زیر محاسبه می شود.

```
m_x, m_y = np.mean(x), np.mean(y)
```

انحراف و انحراف متقابل(cross-deviation) در باره x را به صورت زیر می توان به دست آورد.

```
SS_xy = np.sum(y*x) - n*m_y*m_x
SS_xx = np.sum(x*x) - n*m_x*m_x
```

سپس، ضرایب رگرسیون مانند b را به صورت زیر می توان محاسبه کرد.

```
b_1 = SS_xy / SS_xx
b_0 = m_y - b_1*m_x
return(b_0, b_1)
```

سپس، باید یک تابع تعریف کنیم که به همراه پیش بینی بردار پاسخ، خط رگرسیون را نیز رسم کند.

```
def plot_regression_line(x, y, b):
```

اسکریپت زیر نقاط واقعی را به صورت طرح پراکنده رسم می کند.

```
plt.scatter(x, y, color = "m", marker = "o", s = 30)
```

اسکریپت زیر بردار پاسخ را پیش بینی می کند.

```
y_pred = b[0] + b[1]*x
```

کد زیر خط رگرسیون را رسم خواهد کرد و برچسب ها را روی آنها قرار خواهد داد.

```
plt.plot(x, y_pred, color = "g")
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

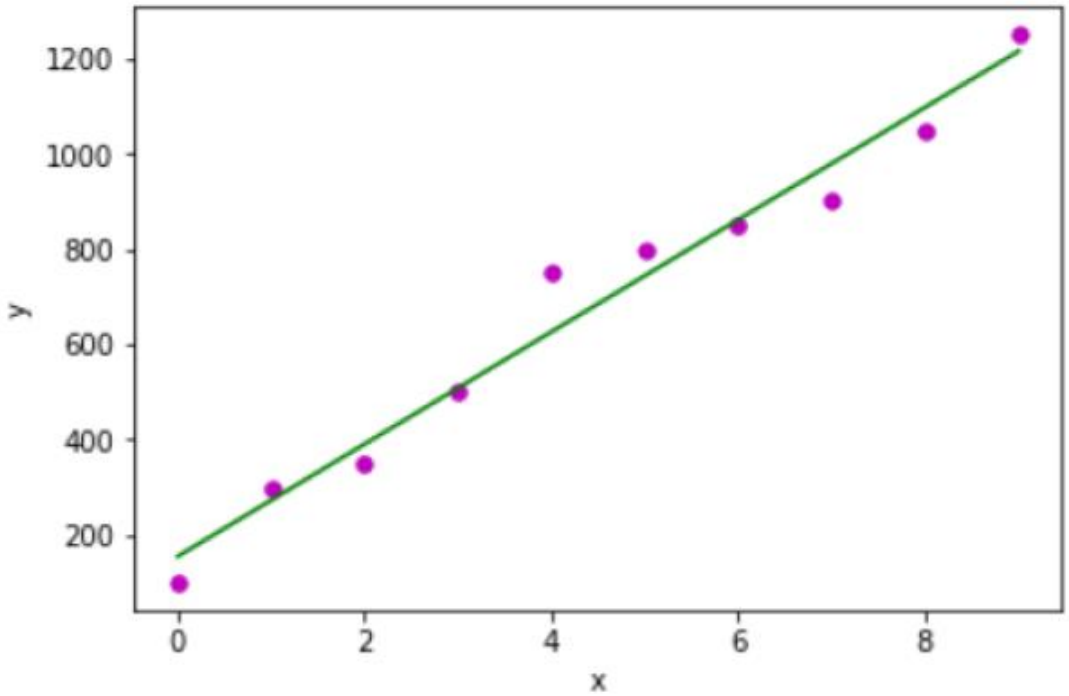
در انتها، برای ارائه مجموعه داده و فراخوانی توابع تعریف شده در بالا، باید تابع main() را تعریف کنیم.

```
def main():
   x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
   y = np.array([100, 300, 350, 500, 750, 800, 850,
900, 1050, 1250])
   b = coef_estimation(x, y)
   print("Estimated coefficients:\nb_0 = {} \nb_1 =
{}".format(b[0], b[1]))
   plot_regression_line(x, y, b)

if __name__ == "__main__":
main()
```

خروجی:

```
Estimated coefficients:
b_0 = 154.5454545454545
b_1 = 117.87878787878788
```



مثال ۲ : در مثال زیر از پیاده سازی پایتون، از مجموعه داده diabetes از scikit-learn استفاده می کنیم.

ابتدا، به صورت زیر با وارد کردن بسته های ضروری شروع می کنیم.

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error,
r2_score
```

سپس، مجموعه داده diabetes را بارگیری می کنیم و شی آن را می سازیم.

```
diabetes = datasets.load_diabetes()
```

از آنجایی که SLR را پیاده سازی می کنیم، فقط از یک ویژگی استفاده خواهیم کرد که در ادامه قرار دارد.

```
X = diabetes.data[:, np.newaxis, 2]
```

سپس، باید به صورت زیر داده را به مجموعه های آموزشی و تست تقسیم کنیم.

```
X_train = X[:-30]
X_test = X[-30:]
```

سپس، باید به صورت زیر هدف را به مجموعه های آموزشی و تست تقسیم کنیم.

```
y_train = diabetes.target[:-30]
```

```
y_test = diabetes.target[-30:]
```

حال، برای آموزش مدل، باید به صورت زیر شی رگرسیون خطی را بسازیم.

```
regr = linear_model.LinearRegression()
```

سپس با استفاده از مجموعه آموزشی، مدل را به صورت زیر آموزش دهید.

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

سپس، به صورت زیر با استفاده از مجموعه تست، پیش بینی ها را انجام دهید.

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

سپس، برخی از ضرایب مانند MSE ، امتیاز انحراف و غیره را به صورت زیر چاپ میکنیم.

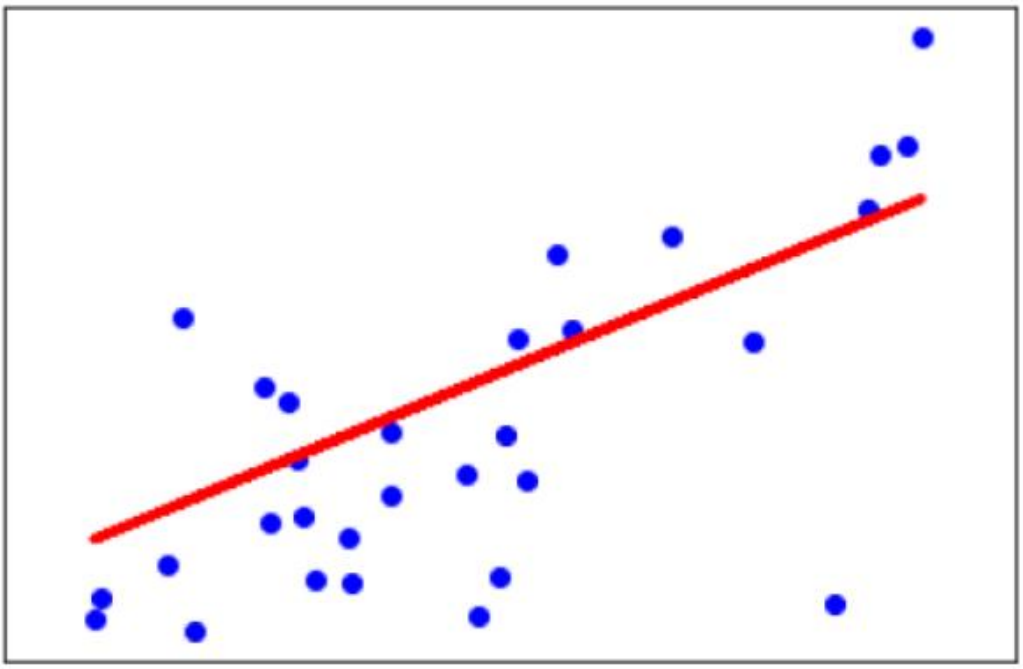
```
print('Coefficients: \n', regr.coef_)
print("Mean squared error: %.2f" %
mean_squared_error(y_test, y_pred))
print('Variance score: %.2f' % r2_score(y_test,
y_pred))
```

حال، خروجی را به صورت زیر رسم می کنیم.

```
plt.scatter(X_test, y_test, color='blue')
plt.plot(X_test, y_pred, color='red', linewidth=3)
plt.xticks(())
plt.yticks(())
plt.show()
```

خروجی:

```
Coefficients:
   [941.43097333]
Mean squared error: 3035.06
Variance score: 0.41
```



رگرسیون خطی چندتایی (MLR): توسعه یافته رگرسیون خطی ساده است که با استفاده از دو ویژگی یا بیشتر، پاسخ را پیش بینی می کند. می توان آن را اینگونه به صورت ریاضی توضیح داد.

یک مجموعه داده که شامل تعداد n مشاهده، تعداد p ویژگی (مانند متغیر های مستقل) و y به عنوان یک پاسخ (مانند متغیر وابسته) باشد را در نظر بگیرید، خط رگرسیون برای ویژگی های p ، به صورت زیر محاسبه می شود.

$$h(xi)=b0+b1xi1+b2xi2+...+bpxip$$

در اینجا، $h(x_i)$ ، مقدار پاسخ پیش بینی شده است و $b_0,b_1,b_2...,b_p$ ضرایب رگرسیون هستند.

مدل های رگرسیون خطی چندتایی همیشه خطاهای موجود در داده که با عنوان خطای باقی مانده شناخته می شوند را در نظر می گیرند، که محاسبات را به صورت زیر تغییر می دهند.

$$h(xi)=b0+b1xi1+b2xi2+...+bpxip+ei$$

همچنین می توان رابطه بالا را به صورت زیر نیز نوشت:

$$yi=h(xi)+eiorei=yi−h(xi)$$

پیاده سازی پایتون: در این مثال، از مجموعه داده Boston housing از scikit learn استفاده خواهیم کرد. ابتدا، با وارد کردن بسته های ضروری به صورت زیر ، شروع می کنیم.

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model, metrics
```

سپس، مجموعه داده را به صورت زیر بارگیری کنید.

```
boston = datasets.load_boston(return_X_y=False)
```

کد زیر ماتریس ویژگی X و بردار پاسخ Y را تعریف می کند.

```
X = boston.data
y = boston.target
```

سپس، به صورت زیر مجموعه داده را به مجموعه های آموزشی و تست تقسیم کنید.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.7, random_state=1)
```

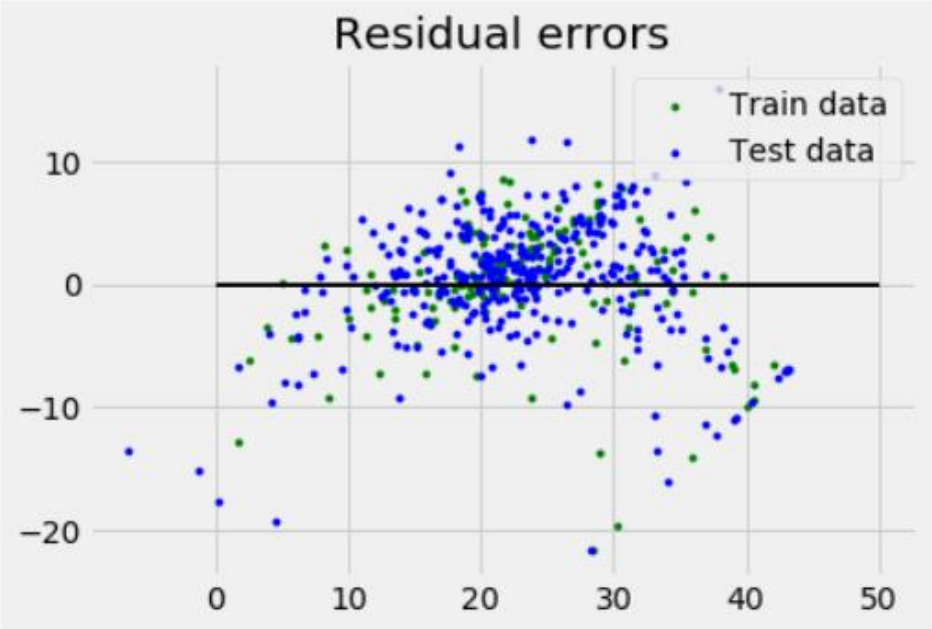
مثال: حال، شی رگرسیون خطی را بسازید و به صورت زیر مدل را آموزش دهید.

```
reg = linear_model.LinearRegression()
reg.fit(X_train, y_train)
print('Coefficients: \n', reg.coef_)
print('Variance  score:  {}'.format(reg.score(X_test,
y_test)))
plt.style.use('fivethirtyeight')
plt.scatter(reg.predict(X_train),
reg.predict(X_train) - y_train,
   color = "green", s = 10, label = 'Train data')
plt.scatter(reg.predict(X_test), reg.predict(X_test) -
y_test,
   color = "blue", s = 10, label = 'Test data')
plt.hlines(y = 0, xmin = 0, xmax = 50, linewidth = 2)
plt.legend(loc = 'upper right')
plt.title("Residual errors")
plt.show()
```

خروجی:

```
Coefficients:
```

```
[
   -1.16358797e-01    6.44549228e-02    1.65416147e-01
1.45101654e+00
   -1.77862563e+01   2.80392779e+00   4.61905315e-02 -
1.13518865e+00
    3.31725870e-01   -1.01196059e-02   -9.94812678e-01
9.18522056e-03
   -7.92395217e-01
]
Variance score: 0.709454060230326
```



فرضیات: موارد زیر برخی از فرضیات مدل رگرسیون خطی درباره مجموعه داده است.

چند همخوانی (Multi-collinearity) : مدل رگرسیون خطی فرض می کند که به مقدار خیلی کم و یا هیچ چند همخوانی در داده وجود ندارد. اساسا، چند همخوانی زمانی رخ می دهد که در متغیر های مستقل و یا خصیصه ها وابستگی وجود داشته باشد.

همبستگی خودکار (Auto-correlation) : از دیگر فرضیات مدل رگرسیون خطی این است که به مقدار خیلی کم و یا هیچ همبستگی خودکار در داده وجود ندارد. اساسا، همبستگی خودکار زمانی رخ می دهد که بین خطا های باقی مانده وابستگی وجود داشته باشد.

ارتباط بین متغیر ها: مدل رگرسیون خطی فرض می کند که رابطه بین متغیر های پاسخ و خصیصه باید خطی باشد.