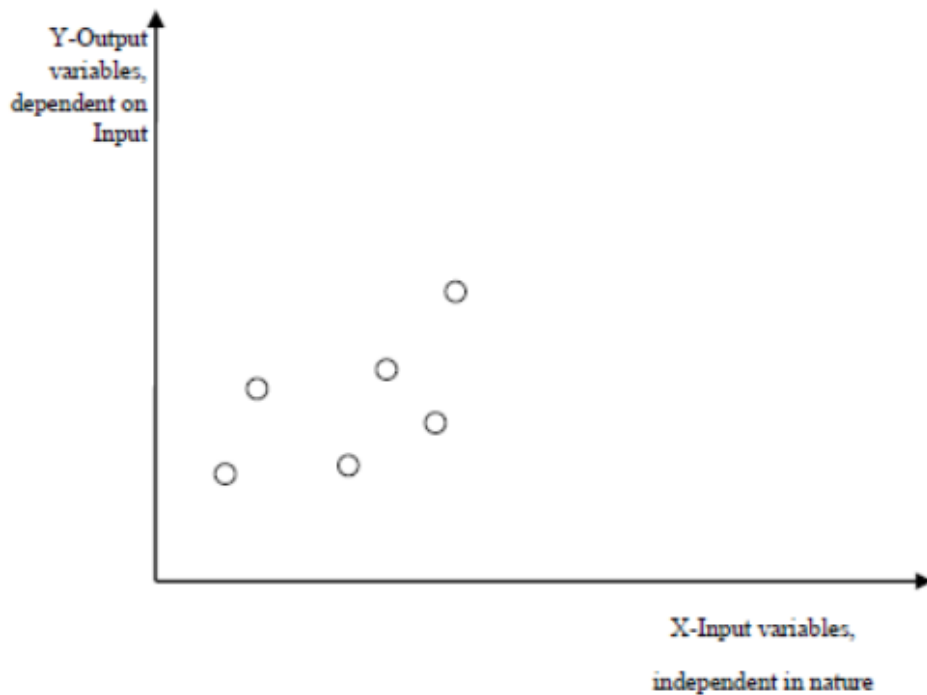
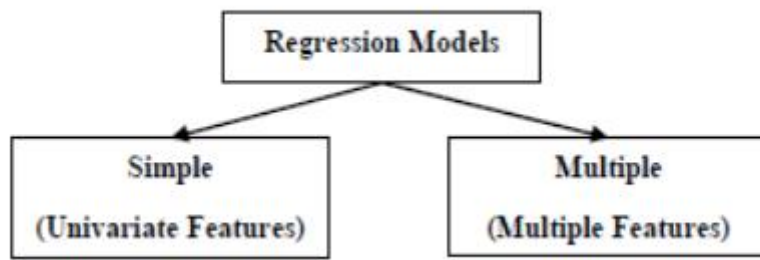


## مقدمه ای بر رگرسیون (Regression) در یادگیری ماشین

رگرسیون یکی دیگر از ابزارهای مهم و رایج آماری و یادگیری ماشین است. هدف اصلی از کارهای مبتنی بر رگرسیون این است که برای داده ورودی فراهم شده، برچسب های خروجی یا پاسخ هایی با مقادیر عددی پیوسته را پیش بینی کند. خروجی بر اساس آن چیزی خواهد بود که مدل در مرحله آموزش، یاد گرفته است. اساساً، مدل های رگرسیون از ویژگی های داده ورودی (متغیرهای مستقل) و مقادیر خروجی عددی پیوسته متناظر با آنها (متغیرهای وابسته یا نتیجه) برای یادگیری وابستگی خاص بین ورودی ها و خروجی های متناظر با آنها استفاده می کنند.



انواع مدل های رگرسیون:



مدل های رگرسیون به دو نوع تقسیم می شوند:

**مدل رگرسیون ساده (Simple regression model):** این ابتدایی ترین مدل رگرسیون است که در آن پیش بینی ها بر اساس یک ویژگی واحد و تک متغیره از داده، صورت می پذیرد.

**مدل رگرسیون چندگانه (Multiple regression model):** همانطور که از نام آن بر می آید، در این مدل از رگرسیون، پیش بینی ها بر اساس چندین ویژگی از داده صورت می پذیرند.

**ساخت یک رگرسر (Regressor) در پایتون:** مدل رگرسر را می توان دقیقا به همان شیوه ساخت طبقه بندی کننده (classifier)، در پایتون ساخت. همچنین می توان از کتابخانه پایتون Scikit-learn، که برای یادگیری ماشین است، برای ساخت رگرسر در پایتون استفاده کرد.

در مثال زیر، یک مدل رگرسیون ابتدایی را خواهیم ساخت که یک خط را متناسب با داده قرار خواهد داد، مثلا رگرسر خطی. مراحل ضروری برای ساخت یک رگرسر در پایتون به شرح زیر است.

**مرحله ۱:** وارد کردن بسته های ضروری پایتون: برای ساخت یک رگرسر با استفاده از scikit-learn، باید

آن را به همراه سایر بسته های لازم، وارد کنیم. با کمک اسکریپت زیر این کار را انجام خواهیم داد.

```

import numpy as np
from sklearn import linear_model
import sklearn.metrics as sm
import matplotlib.pyplot as plt
  
```

**مرحله ۲:** پس از وارد کردن بسته های ضروری، به یک مجموعه داده برای ساخت مدل پیش بینی رگرسیون نیاز داریم. می توانیم آن را از مجموعه داده `sklearn` وارد کنیم یا بر اساس نیاز خود از سایر مجموعه های داده استفاده کنیم. ما از داده ذخیره شده خود استفاده خواهیم کرد. با کمک اسکریپت زیر آن را وارد می کنیم.

```
input = r'C:\linear.txt'
```

سپس، برای بارگیری این داده از تابع `np.loadtxt` استفاده خواهیم کرد.

```
input_data = np.loadtxt(input, delimiter=',')
X, y = input_data[:, :-1], input_data[:, -1]
```

**مرحله ۳:** سازماندهی داده به مجموعه های آموزشی و تست: از آنجایی که باید مدل خود را با داده های دیده نشده تست کنیم، مجموعه داده خود را به دو بخش مجموعه آموزشی و مجموعه تست تقسیم خواهیم کرد. دستورات زیر این کار را انجام خواهند داد.

```
training_samples = int(0.6 * len(X))
testing_samples = len(X) - num_training
X_train, y_train = X[:training_samples],
y[:training_samples]
X_test, y_test = X[training_samples:],
y[training_samples:]
```

**مرحله ۴:** ارزیابی و پیش بینی مدل: پس از تقسیم داده به دو بخش آموزشی و تست، باید مدل را بسازیم.

برای این هدف از تابع `LinearRegression()` مربوط به `Scikit-learn` استفاده می کنیم. دستورات زیر یک شی رگرسیون خطی را خواهند ساخت.

```
reg_linear = linear_model.LinearRegression()
```

سپس، این مدل را به صورت زیر با نمونه های آموزشی، آموزش دهید.

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

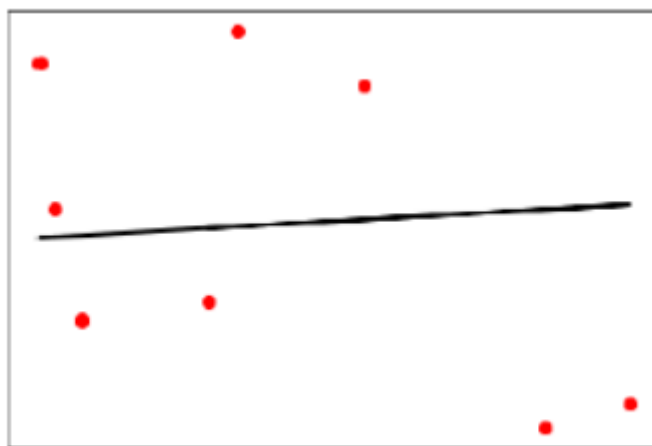
حال، در انتها باید با داده تست، پیش بینی کنیم.

```
y_test_pred = reg_linear.predict(X_test)
```

**مرحله ۵:** رسم و تصویر سازی: پس از پیش بینی، با کمک اسکریپت زیر می توانیم آن را رسم کنیم و به تصویر بکشیم.

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_test, y_test_pred, color = 'black',
linewidth = 2)
plt.xticks(())
plt.yticks(())
plt.show()
```

**خروجی:**



در خروجی فوق، می توانیم خط رگرسیون را بین نقاط داده ببینیم.

**مرحله ۶:** محاسبه کارایی: همچنین می توانیم همانطور که در ادامه آمده است، با کمک معیار های متنوع کارایی، کارایی مدل رگرسیون خود را محاسبه کنیم.

```
print("Regressor model performance:")
print("Mean absolute error (MAE) =",
round(sm.mean_absolute_error(y_test, y_test_pred), 2))
print("Mean squared error (MSE) =",
round(sm.mean_squared_error(y_test, y_test_pred), 2))
print("Median absolute error =",
round(sm.median_absolute_error(y_test, y_test_pred),
2))
```

```
print("Explain variance score =",  
      round(sm.explained_variance_score(y_test,  
y_test_pred), 2))  
print("R2 score =", round(sm.r2_score(y_test,  
y_test_pred), 2))
```

**خروجی:**

```
Regressor model performance:  
Mean absolute error(MAE) = 1.78  
Mean squared error(MSE) = 3.89  
Median absolute error = 2.01  
Explain variance score = -0.09  
R2 score = -0.09
```

**انواع الگوریتم های رگرسیون ML:** مفیدترین و محبوب ترین الگوریتم رگرسیون، الگوریتم رگرسیون

خطی است که در ادامه به دو نوع الگوریتم رگرسیون خطی ساده و الگوریتم رگرسیون خطی چندتایی تقسیم

می شود. در بخش بعدی در این باره صحبت خواهیم کرد و آن را در پایتون پیاده سازی خواهیم کرد.

برنامه های کاربردی: برنامه های کاربردی مربوط به الگوریتم های رگرسیون ML در ادامه معرفی شده است.

**پیش بینی یا تحلیل پیشگویانه:**

یکی از مهمترین کاربرد های رگرسیون پیش بینی یا تحلیل پیشگویانه است. برای مثال، می توانیم GDP ،

قیمت نفت یا به بیان ساده تر، داده کمی که با گذر زمان تغییر می کند را پیش بینی کنیم.

**بهینه سازی:** می توان با کمک رگرسیون پردازش کسب و کار و تجارت را بهینه کرد. برای مثال، یک مدیر

فروشگاه می تواند با ساخت یک مدل آماری، زمان اوج ورود مشتریان را بفهمد.

**تصحیح خطا:** در تجارت، تصمیم گیری درست به اندازه بهینه سازی پردازش کسب و کار دارای اهمیت

است. رگرسیون به ما کمک می کند تصمیمات درستی بگیریم همانطور که در تصحیح تصمیمات پیاده سازی

شده کمک می کند.

**اقتصاد:** رگرسیون کاربردی ترین ابزار در اقتصاد است. از آن می توان برای پیش بینی عرضه، تقاضا، مصرف و فهرست سرمایه گذاری استفاده کرد.

**امور مالی:** یک شرکت مالی همیشه علاقه مند به کمینه کردن سبد ریسک است و می خواهد از عوامل تاثیر گذار بر مشتریان مطلع باشد. با کمک مدل رگرسیون می توان تمامی این موارد را پیش بینی کرد.

