جامعة حمص

الكلية التطبيقية

قسم تقنيات حاسوب

السنة الثالثة

قواعد معطيات٣

القسم العملي

المحاضرة الخامسة

اعداد المدرسين:

م. زينب مراد **م. بتول الليوس**

الانحدار(Regression)

What is Regression:

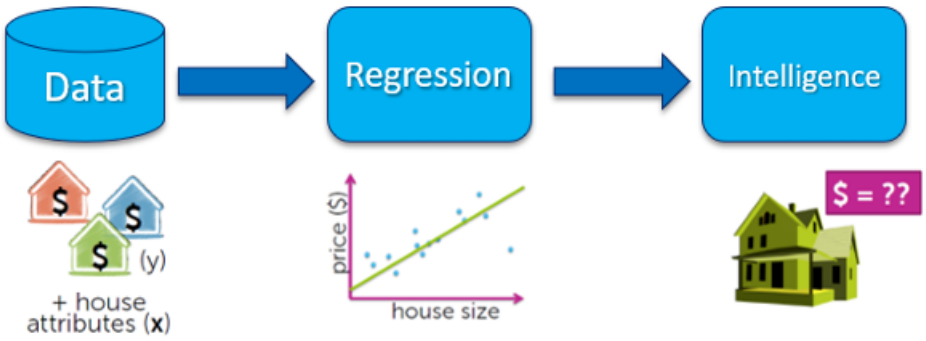
- ✓ Used to predict a **continuous** "output" (**y**) based on **set of features** (**X**) derived from data.
- ✓ in order to make predictions, regression tries to learn the relationship between the inputs (**X**) and the output (**y**).
- ✓ For example, we can use regression for estimating a house price based on attributes like (Size, #bedrooms, #bathrooms, Sq.ft. living, and many others…).
- ✓ Many regression models: simple regression vs. multiple regression.

الانحدار هو أسلوب إحصائي يُستخدم للتنبؤ بقيمة مستمرة لمتغير(مثل سعر المنزل) بناءً على متغيرات أخرى أو ميزات (مثل المساحة وعدد الغرف) الهدف هو فهم العلاقة بين المدخلات ؟ (X) والمخرجات(y)

- يوجد أنواع متعددة من نماذج الانحدار، مثل:

الانحدار البسيط (Simple Regression)

الانحدار المتعدد (Multiple Regression)



مثال: التنبؤ بأسعار بيع المنازل

- يمكننا استخدام الانحدار لتقدير سعر المنزل بناءً على خصائص مثل:

-المساحة

-عدد الحمامات

- عدد غرف النوم

- مساحة المعيشة (sqft_living)

وغيرها

الانحدار البسيط (Simple Linear Regression)

✓ Simple Regression:

- The simplest model we can use Simple Linear Regression by just fitting **a line** to our data points (i.e., between the input and the output).
- Practically, the simple linear model is defined in terms of a set of parameters: **w0** (intercept) and **w1** (slope).
- For a given training dataset, there will be multiple lines and each one is given by a different set of parameters $\widehat{W}$.
    a. which line is the best to use?
    b. Which $\widehat{W}$ do we have to choose for our model?
- To measure the cost (**quality**) associated with a specific fit (line), we need a quality metric e.g., residual sum of squares (RSS).

هو أبسط نموذج، حيث نقوم بتمثيل العلاقة بخط مستقيم بين المدخلات والمخرجات

يتم تحديد هذا الخط باستخدام: نقطة التقاطع والميل

w_0 : نقطة التقاطع (intercept)

w_1 : الميل أو الانحدار (slope)

- كل خط محتمل له مجموعة مختلفة من القيم w_1 و w_0

والسؤال هو أي مجموعة من القيم (W) نختارها؟ أي خط هو الأفضل؟

*** لحساب جودة الخط، نستخدم مقياس مثل:**

(RSS): هو اختصار لـ Residual Sum of Squares أي مجموع مربعات البواقي

وهو يقيس مدى قرب الخط من النقاط الفعلية .

خطوات التنبؤ بسعر المنزل باستخدام Python

أولا" : استيراد المكتبات

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

*** لبناء نماذج الانحدار باستخدام مكتبة scikit-learn (sklearn) نكتب :**

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import meansquarederror
```

ثانيا" : تحميل البيانات (قراءة البيانات من ملف)

housesalespricesdata = pd.readcsv ('housesalesprices.csv')

housesalesprices_data.head()

يعرض أول صفوف من البيانات، والتي تحتوي على معلومات مثل:

هل يوجد إطلالة على الماء؟ -عدد الطوابق -مساحة المعيشة والأرض -عدد غرف النوم والحمامات -السعر -التاريخ -

سنة البناء -التقييم العام للمنزل

```
#Load our dataset:
house_sales_prices_data = pd.read_csv('house_sales_prices.csv')
#Navigating through our dataset:
house_sales_prices_data.head() #view first rows of our data
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_built
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0	0	0	...	7	1180	0	1955
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0	0	0	...	7	2170	400	1951
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0	0	0	...	6	770	0	1933
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0	0	0	...	7	1050	910	1965
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0	0	0	...	8	1680	0	1987

ثالثا" : وصف البيانات إحصائيًا

housesalesprices_data.describe()

يعرض معلومات مثل: القيم الدنيا والقصوى -الانحراف المعياري -المتوسط - الربع الأول والثالث

```
house_sales_prices_data.describe()
```

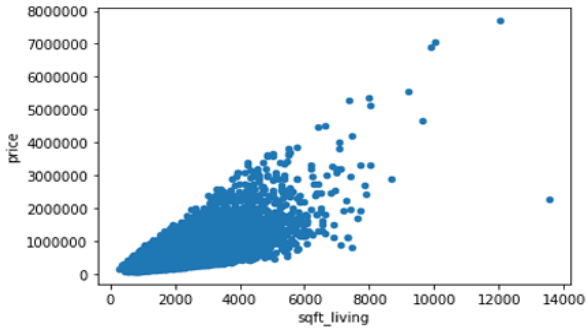
	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000
mean	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.494309	0.007542	0.234303	3.409430
std	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	0.539989	0.086517	0.766318	0.650743
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000	0.000000	0.000000	1.000000
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000	0.000000	0.000000	3.000000
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000	0.000000	0.000000	3.000000
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000	0.000000	0.000000	4.000000
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000	1.000000	4.000000	5.000000

رابعا" : رسم العلاقة بين المساحة و السعر

housesalespricesdata.plot.scatter(x='sqftliving', y='price')

يرسم نقاط تمثل العلاقة بين مساحة المعيشة (x) وسعر المنزل(y)، مما يساعدنا على رؤية الاتجاه العام

```
house_sales_prices_data.plot.scatter(x='sqft_living', y='price') #draw our data points here
we have x-axis (sqft_livinig) and y-axis (price)
```



ملاحظة: بالنظر إلى المساحة السكنية (بالقدم المربع) للمنازل في مجموعة الاختبار، يتم استخدام طريقة التنبؤ لتقدير أو توقع قيمة المنزل (السعر) استنادًا إلى النموذج الملائم

Feature Selection اختيار الميزات

نبدأ بتحديد الأعمدة التي نريد استخدامها كمدخلات (X) والناتج الذي نريد التنبؤ به(y)

```
X = housesalesprices_data.iloc[:, [5]].values          # نختار العمود رقم ٥ (مساحة المعيشة)
y = housesalesprices_data.iloc[:, 2].values            # نختار العمود رقم ٢ (سعر المنزل)
```

نستخدم الأقواس المربعة المزدوجة [[5]] لأن مكتبةscikit-learn تتطلب أن تكونX مصفوفة ثنائية الأبعاد

بناء نموذج الانحدار الخطي Linear Regression

1 - تهيئة النموذج :

```
np.random.seed(39)
reg = LinearRegression()
```

2 - تدريب النموذج على البيانات:

```
reg.fit(X, y)
```

3 - رسم العلاقة بين المدخلات والنواتج :

```
plt.plot(X, y, '.', X, reg.predict(X), '-')
plt.title(' Real train value VS predicted values using our model ')
plt.xlabel(' Sq. ft. living of the house')
plt.ylabel(' House Price')
```

4 - طباعة معاملات النموذج :

```
reg.coef_          # معامل الانحدار (ميل الخط)
reg.intercept_     # نقطة التقاطع مع المحورy
```

معامل الانحدار: ٢٨٠,٦٢ - نقطة التقاطع: -٤٣٥٨٠,٧٤ -

5 - تقييم النموذج باستخدام متوسط مربع الخطأ :

meansquarederror(y, reg.predict(X))

I. Now we'll going to use LinearRegression class in scikit-learn library to build our model.

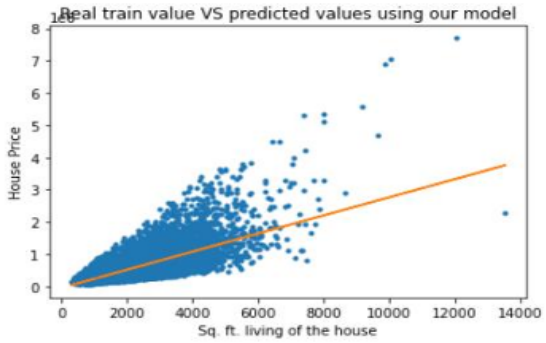
```
np.random.seed(39)
reg = LinearRegression()  #initialize LinearRegression Class.
```

II. We have a method called fit which takes our x and y from training set to model the relationship between sq. ft. living and price.

```
reg.fit(X,y)  # fit the model to our training data.
```

III. Then we'll plot this fitted line to our data and our test set points to see how our line fitted the data:

```
plt.plot(X, y,'.',X,reg.predict(X),'-')
plt.title('Real train value VS predicted values using our model')
plt.ylabel('House Price')
plt.xlabel('Sq. ft. living of the house')
```



IV. Finally, we're going to print the coefficients of the fitted line:

```
reg.coef_
reg.intercept_
```

```
reg.coef_
array([280.62356663])

reg.intercept_
-43580.740327085136
```

V. Evalute the generated model:

```
mean_squared_error(y,reg.predict(X))
```

Multiple Regression الانحدار المتعدد

Multiple Regression is linear regression with multiple features. We have two cases:

- We have a single input and we're trying to use more complex functions of that single input to predict the output.
- We have multiple inputs (different attributes that used to build our regression model).

in previous example we just focus on sq. ft. living room to predict our house price, but as we mentioned before we may have multiple attributes that we can add to our regression model. we'll add another attribute such number of bedrooms with number of bathrooms to predict our house price.

نستخدم أكثر من ميزة (مثل عدد الغرف وعدد الحمامات) لتحسين التنبؤ بسعر المنزل

هذا النموذج يأخذ أكثر من دخل لتوقع السعر بدقة أعلى

مثال:

```
X = housesalesprices_data.values [['bedrooms', 'bathrooms']]

y = housesalesprices_data.values ['price']

reg.fit(X, y)
```

خطوات تنفيذ الانحدار المتعدد في Python

1- استيراد المكتبات الضرورية

```
import pandas as pd                                    # معالجة البيانات
import numpy as np                                     # معالجة البيانات
import matplotlib.pyplot as plt                        # لرسم المخططات ثلاثية الأبعاد
from mpl_toolkits.mplot3d import Axes3D                # لرسم المخططات ثلاثية الأبعاد
from sklearn.linear_model import LinearRegression      # لبناء النموذج
from sklearn.metrics import meansquarederror           # لتقييم دقة النموذج
```

2 - قراءة البيانات واختيار الخصائص

```
housesalespricesdata = pd.readcsv('housesalesprices.csv')      # قراءة ملف البيانات
X = housesalesprices_data[['bathrooms', 'bedrooms']]           # يحتوي على عدد الحمامات وغرف النوم
y = housesalesprices_data.price                                # يحتوي على أسعار المنازل (القيمة المستهدفة )
```

3 - بناء النموذج واستخراج المعاملات

```
reg = LinearRegression()
reg.fit(X, y)
a1, a2 = reg.coef_                    # معاملات التأثير لكل من الحمامات وغرف النوم
b = reg.intercept_                    # الثابت (الانحراف)
```

الناتج :

```
a1,a2 = reg.coef_
print(a1,"\t",a2)
```

```
237780.58423560805      20138.265844201756
```

```
b = reg.intercept_
b
```

```
-30642.994356740615
```

تأثير كل حمام إضافي ← a1 = 237788.58

تأثير كل غرفة نوم إضافية ← a2 = 20138.27

الثابت ← b = -30642.99

4 - رسم السطح التنبؤي ثلاثي الأبعاد

يتم إنشاء شبكة من القيم المحتملة للحمامات وغرف النوم

يتم حساب السعر المتوقع باستخدام المعادلة

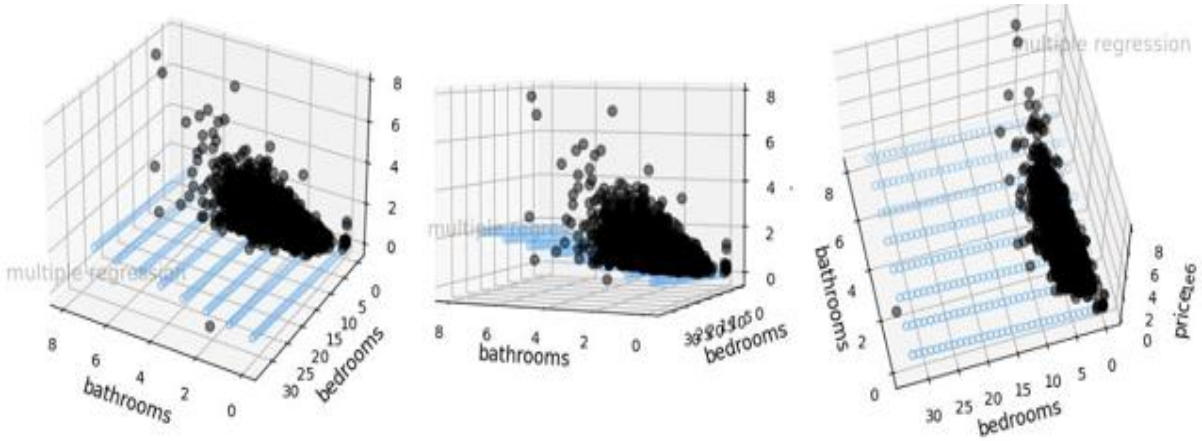
```
Z = a1 * X1 + a2 * Y + b
```

يتم رسم النقاط الأصلية والسطح التنبؤي في ثلاث زوايا مختلفة

5- تقييم النموذج

```
meansquarederror(y, reg.predict(X))
```

هذا يعطيك متوسط مربع الخطأ، وهو مقياس لدقة النموذج: كلما كان أقل، كان النموذج أفضل



- **Import Required Libraries:**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D #to plot 3D Charts.
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

- **Read our dataset and select features:**

```
house_sales_prices_data = pd.read_csv('house_sales_prices.csv') #load our dataset
X = house_sales_prices_data[['bathrooms', 'bedrooms']] #select the input attributes
y = house_sales_prices_data.price #select target attribute
np.random.seed(39)
```

- **Build our model and print its coefficients:**

```
reg = LinearRegression()
reg.fit(X,y)
a1, a2 = reg.coef_
b = reg.intercept_
```

```
a1,a2 = reg.coef_
print(a1,"\t",a2)
```

```
237780.58423560805      20138.265844201756
```

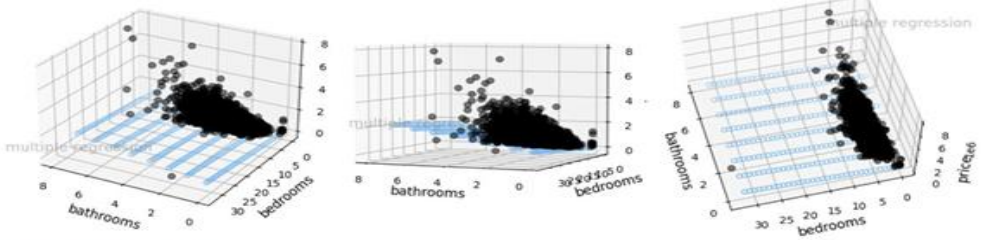
```
b = reg.intercept_
b
```

```
-30642.994356740615
```

- **Plot the fitted surface:**

```
plt.style.use('default')

fig = plt.figure(figsize=(12, 4))

ax1 = fig.add_subplot(131, projection='3d')
ax2 = fig.add_subplot(132, projection='3d')
ax3 = fig.add_subplot(133, projection='3d')

axes = [ax1, ax2, ax3]
```

```
x,y1,z=np.array(house_sales_prices_data['bathrooms']),np.array(house_sales_prices_data['bedro
oms']), np.array(y)
X1, Y = np.meshgrid(np.arange(0, 8, 1), np.arange(0, 33, 1)) #generate the surface space
Z = a1 * X1 + a2 * Y + b
for ax in axes:
    ax.plot(x, y1, z, color='k', zorder=15, linestyle='none', marker='o', alpha=0.5)
    ax.scatter(X1.flatten(), Y.flatten(), Z, facecolor=(0,0,0,0), s=20, edgecolor='#70b3f0')
    #Flatten() function is used to return one dimension array
    ax.set_xlabel('bathrooms', fontsize=12)
    ax.set_ylabel('bedrooms', fontsize=12)
    ax.set_zlabel('price', fontsize=12)

ax1.text2D(0.2, 0.32, 'multiple regression', fontsize=13, ha='center', va='center',
           transform=ax1.transAxes, color='grey', alpha=0.5)
ax2.text2D(0.3, 0.42, 'multiple regression', fontsize=13, ha='center', va='center',
           transform=ax2.transAxes, color='grey', alpha=0.5)
ax3.text2D(0.85, 0.85, 'multiple regression', fontsize=13, ha='center', va='center',
           transform=ax3.transAxes, color='grey', alpha=0.5)

ax1.view_init(elev=28, azim=120)
ax2.view_init(elev=4, azim=114)
ax3.view_init(elev=60, azim=165)
fig.tight_layout()
```




```
#Evalute the fitted model:
mean_squared_error(y,reg.predict(X))
```

References:

1- **Pandas docs [https://pandas.pydata.org/docs/].**
2- **NumPy docs [https://numpy.org/doc/].**
3- **Matplotlib docs [https://matplotlib.org/3.3.3/contents.html].**
4- **SciKit-learn docs [https://scikit-learn.org/0.21/documentation.html].**
5- **https://aegis4048.github.io/mutiple_linear_regression_and_visualization_in_python**
6- **https://linuxtut.com/en/04563d8f1572290e99b3/**