

# پروژه پایتون اقتصاد مهندسی

اعضا:

آرش اسلامی 401103871

محمد رضا روف 401104135

محمد صابر مهدوی 401104481

امیرپارسا جهانبانی فرد 401103977

سید ارشیا شاهرخی 401170423

محمد طاها عسگری 401104287

کتابخانه پانداس و نامپای را فراخوانی می‌کنیم.

```
] : import pandas as pd
import numpy as np
from func import *
```

دیتا های داخل اکسل را فراخوانی می‌کنیم.

```
] : df = pd.read_excel('data.xlsx')
df
```

```
] :
```

	Years	Salary	material 1	material 2	demand	price
0	1990	39343.0	200.000000	89.600000	103	1200.0
1	1991	46205.0	212.000000	92.662500	104	1375.0
2	1992	37731.0	224.720000	95.807250	101	1451.0
3	1993	43525.0	238.203200	99.034819	99	1593.0
4	1994	39891.0	252.495392	102.345626	95	1718.5
5	1995	56642.0	267.645116	105.739927	96	1844.0
6	1996	60150.0	283.703822	109.217795	95	1969.5
7	1997	54445.0	300.726052	112.779099	92	2095.0
8	1998	64445.0	318.769615	116.423489	92	2220.5

شرح تمرین :

1) دیتا های خالی را پیدا میکنیم.

```
] : df.isnull().sum()
```

مقادیر خالی سه مقدار در بخش حقوق کارکنان بود این مقادیر را در جدول با میانگین مقادیر قبل و بعد پر کردیم و ثبت نهایی کردیم.

```
] : def fill_nan(df,column):  
    for i in range(1,len(df)-1):  
        if pd.isna(df.loc[i,column]):  
            uper_cell = df.loc[i - 1,column]  
            lower_cell = df.loc[i + 1,column]  
            if pd.notna(uper_cell) and pd.notna(lower_cell):  
                df.loc[i, column] = (uper_cell + lower_cell) / 2  
fill_nan(df, 'Salary')  
df.to_excel('data.xlsx', index=False)  
df
```

در اینجا ما تابعی تعریف کردیم که مقادیر خالی در ستون مد نظر را پیدا کند و در صورت پر بودن مقادیر سال قبل و سال بعدش میانگین این مقادیر را در آن قرار می‌دهد و در اکسل اصلی این مقادیر را نهایی می‌کند. مشاهده می‌شود که مقادیر خالی مان صفر شده اند این نکته مهم است که با توجه به اینکه در اکسل ثبت نهایی کردیم این مقادیر خالی را در هر دو جدول مقادیر خالی همواره مقادیر صفر است.

```
] : df.isnull().sum()
```

```
] : Years          0  
Salary            0  
material 1        0  
material 2        0  
demand            0  
price             0  
dtype: int64
```

تابعی برای بدست آوردن رگرسیون مینویسیم به این صورت که ابتدا مقادیر طولی و عرضی آن را مشخص می‌کنیم و آنها رو مدل می‌کنیم و این مدل را فیت می‌کنیم و با استفاده از خاصیت کتابخانه عرض از مبدا و شیب را بدست آورده و آنها را با این تابع برمی‌گردانیم.

```
] : def regresion(x,y):  
    X = x.values.reshape(-1,1)  
    Y = y.values.reshape(-1,1)  
    model = LinearRegression()  
    model.fit(X,Y)  
    intercept = model.intercept_[0]  
    coefficient = model.coef_[0][0]  
    return intercept ,coefficient
```

تابعی برای پیش‌بینی مقادیر نوشته ایم که در آن از تابع رگرسیون نیز استفاده کردیم.

```
def predict(x,y,t):  
    z = regresion(y,t)  
    return z[0] + z[1] * x
```

تابعی برای دقت رگرسیون تعریف کردیم که مانند تابع رگرسیون است با فرق اینکه از پیش فرض دیگر کتابخانه در آن استفاده کردیم.

```
def score(x,y):  
    X = x.values.reshape(-1,1)  
    Y = y.values.reshape(-1,1)  
    model = LinearRegression()  
    model.fit(X,Y)  
    score = model.score(X,Y)  
    return score
```

2) مقادیر رگرسیون (عرض از مبدا و شیب) را با استفاده از تابعی که بالا تعریف کردیم بدست آوردیم.

```
regresion(df['Years'],df['Salary'])
```

```
(-5249679.051136364, 2655.052640374332)
```

پیش بینی را تا سال 2050 ادامه دادیم.

```
predicted_list=[]  
for i in range(2023,2051):  
    predicted_list.append((i,predict(i,df['Years'],df['Salary'])))  
predicted_list
```

```
[(2023, 121492.44034090918),  
(2024, 124147.49298128393),  
(2025, 126802.54562165774),  
(2026, 129457.5982620325),  
(2027, 132112.6509024063),  
(2028, 134767.70354278106),  
(2029, 137422.7561831558),  
(2030, 140077.80882352963),  
(2031, 142732.86146390438),  
(2032, 145387.9141042782),  
(2033, 148042.96674465295),  
(2034, 150698.01938502677),  
(2035, 153353.07202540152),  
(2036, 156008.12466577534),  
(2037, 158663.1773061501),  
(2038, 161318.2299465239),  
(2039, 163973.28258689865),  
(2040, 166628.3352272734),  
(2041, 169283.38786764722),  
(2042, 171938.44050802197),  
(2043, 174593.4931483958),  
(2044, 177248.54578877054),  
(2045, 179903.59842914436),  
(2046, 182558.6510695191),  
(2047, 185213.70370989293),  
(2048, 187868.75635026768),  
(2049, 190523.80899064243),  
(2050, 193178.86163101625)]
```

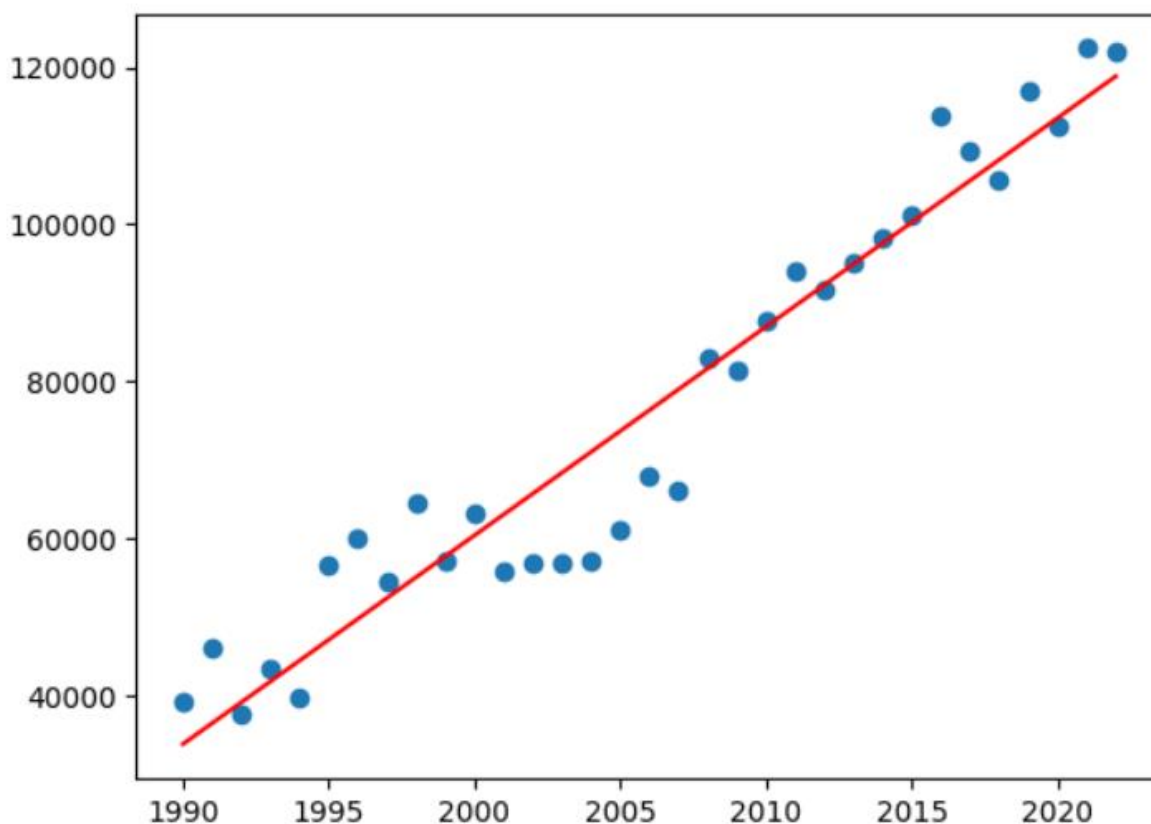
کتابخانه متپلات را برای رسم نمودار ها فراخوانی کردیم.

تابعی تعریف کردیم که نمودار نقطه ای مقادیر و رگرسیون خطی را در آن رسم کنیم.

```
def pltreg(m,n):  
    plt.scatter(m,n)  
    X = m.values.reshape(-1,1)  
    Y = n.values.reshape(-1,1)  
    model = LinearRegression()  
    model.fit(X,Y)  
    plt.plot(m,model.predict(X),c = 'red')  
    plt.show()
```

3)نمودار حقوق بر اساس سال:

```
pltreg(df['Years'],df['Salary'])
```



دقت رگرسیون را با استفاده از تابع تعریف شده بدست میاریم.

```
score(df['Years'],df['Salary'])
```

```
0.9312739842698768
```

با توجه به دقت رگرسیون بدست آمده که تقریباً 93٪ است نشان دهنده این ماجرا است که شیب حقوق در طی این زمان همواره مثبت بوده و تغییرات کمی که باعث شده این دقت به یک نرسد شرایط اقتصادی مختلف و جهانی و محیطی است که اتفاق افتاده که کمی افول و صعود داشته است.

تابعی تعریف کردیم برای بدست آوردن ارزش آینده مقادیر (تابع P به F)

```
def POM1(i,n,P):  
    F = P * (1 + i) ** n  
    return F
```

4) قیمت متریال 1 تا سال 2050 با استفاده از تابع تعریف شده.

```
i = 0.05 + 0.01  
for m in range(2023,2051):  
    n=m-df.iloc[0,0]  
    p = df.iloc[0,2]  
    print(f'{m} : {POM1(i,n,p)}')
```

```
2023 : 1368.1179765596917  
2024 : 1450.2050551532734  
2025 : 1537.21735846247  
2026 : 1629.4503999702179  
2027 : 1727.2174239684311  
2028 : 1830.8504694065373  
2029 : 1940.7014975709294  
2030 : 2057.1435874251856  
2031 : 2180.5722026706967  
2032 : 2311.4065348309387  
2033 : 2450.090926920795  
2034 : 2597.0963825360427  
2035 : 2752.922165488206  
2036 : 2918.097495417498  
2037 : 3093.1833451425478  
2038 : 3278.774345851101  
2039 : 3475.5008066021674  
2040 : 3684.030854998298  
2041 : 3905.0727062981955  
2042 : 4139.377068676087  
2043 : 4387.739692796653  
2044 : 4651.004074364452  
2045 : 4930.06431882632  
2046 : 5225.868177955899  
2047 : 5539.420268633254  
2048 : 5871.785484751249  
2049 : 6224.092613836324  
2050 : 6597.538170666505
```

این قیمت از یک نسبت خاص 6 درصدی بدست آمده (پیش بینی) است. زیرا تاثیراتی که بر این قیمت در سال‌ها بوده است تنها تورم و نرخ بهره است.

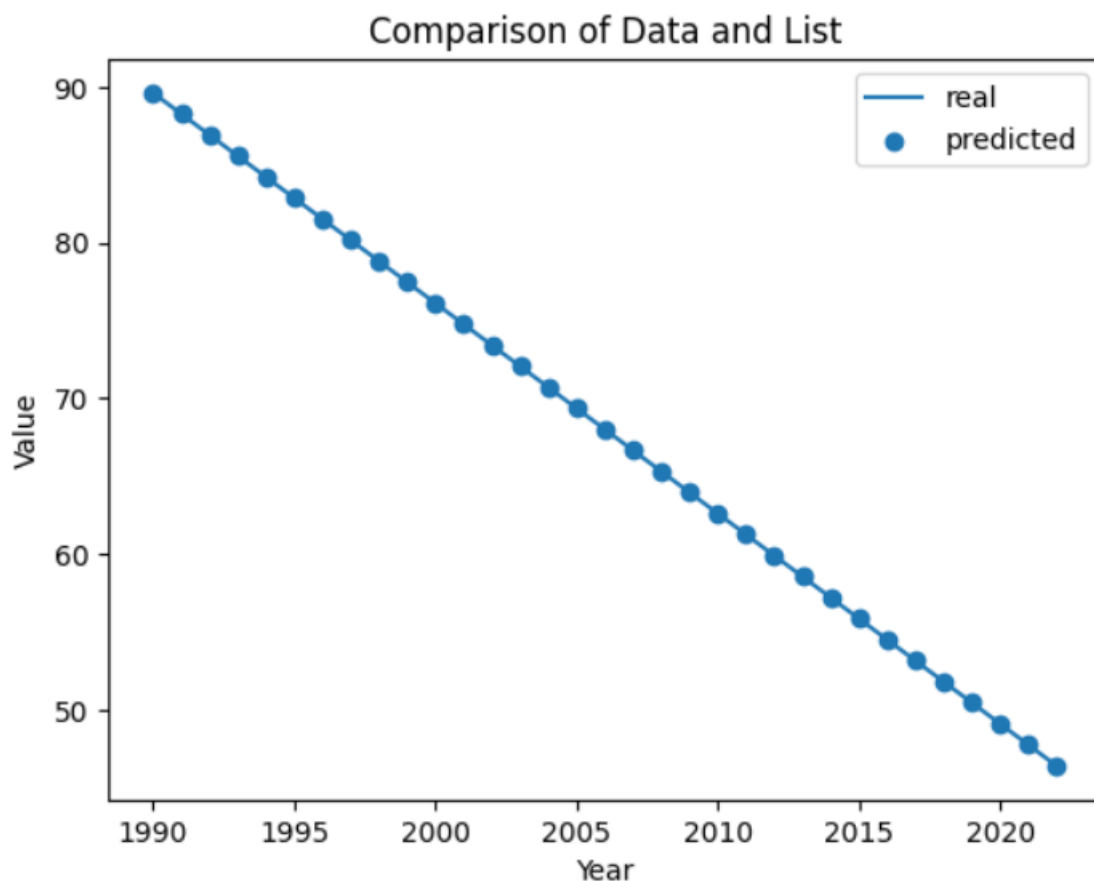
5) در ابتدا از آنجا که از کلمه ارزش استفاده شده ، تمامی داده ها را به سال 1990 با نرخ بهره 5 درصد میبریم. این کار را با استفاده از توابع تعریف شده که در فایل `func.py` وجود دارد انجام میدهیم. سپس تابع گفته شده را بر روی ستون متریکال گفته شده اعمال کرده و ستون جدیدی ایجاد میکنیم.

داده های جدید را در ستون `p` ذخیره میکنیم.

	Years	Salary	material 1	material 2	demand	price	n	P
0	1990	39343.0	200.000000	89.600000	103	1200.0	0	89.60
1	1991	46205.0	212.000000	92.662500	104	1375.0	1	88.25
2	1992	37731.0	224.720000	95.807250	101	1451.0	2	86.90
3	1993	43525.0	238.203200	99.034819	99	1593.0	3	85.55
4	1994	39891.0	252.495392	102.345626	95	1718.5	4	84.20
5	1995	56642.0	267.645116	105.739927	96	1844.0	5	82.85
6	1996	60150.0	283.703822	109.217795	95	1969.5	6	81.50
7	1997	54445.0	300.726052	112.779099	92	2095.0	7	80.15
8	1998	64445.0	318.769615	116.423489	92	2220.5	8	78.80
9	1999	57189.0	337.895792	120.150370	87	2346.0	9	77.45
10	2000	63218.0	358.169539	123.958881	84	2471.5	10	76.10
11	2001	55794.0	379.659712	127.847867	86	2597.0	11	74.75
12	2002	56957.0	402.439294	131.815854	87	2722.5	12	73.40
13	2003	57019.0	426.585652	135.861021	78	2848.0	13	72.05
14	2004	57081.0	452.180791	139.981164	79	2973.5	14	70.70
15	2005	61111.0	479.311639	144.173669	78	3099.0	15	69.35
16	2006	67938.0	508.070337	148.435472	78	3224.5	16	68.00
17	2007	66029.0	538.554557	152.763021	71	3350.0	17	66.65
18	2008	83088.0	570.867831	157.152236	73	3475.5	18	65.30
19	2009	81363.0	605.119900	161.598465	72	3601.0	19	63.95
20	2010	87651.5	641.427094	166.096436	71	3726.5	20	62.60
21	2011	93940.0	679.912720	170.640209	67	3852.0	21	61.25
22	2012	91738.0	720.707483	175.223117	70	3977.5	22	59.90
23	2013	95005.5	763.949932	179.837716	69	4103.0	23	58.55
...								
29	2019	116969.0	1083.677580	207.659041	62	4856.0	29	50.45
30	2020	112635.0	1148.698235	212.207371	59	4981.5	30	49.10
31	2021	122391.0	1217.620129	216.691386	58	5107.0	31	47.75
32	2022	121872.0	1290.677336	221.093284	57	5232.5	32	46.40



(6) سپس طبق داده های بدست آمده مدل رگرسیون خود را فیت میکنیم



```
from sklearn.metrics import r2_score  
  
r2 = r2_score(value, y)  
print("R-squared: ", r2)
```

R-squared: 1.0

مشاهده میکنیم که داده های پیشبینی شده و داده های بدست آمده با تقریب خوبی کاملاً مطابق هم هستند و میتوان داده ها را خطی تقریب زد

(7) تابع خواسته شده را تعریف میکنیم. سپس طبق آن برای هر سال مقدار شاخص گفته شده را بدست میآوریم. حال برای فیت کردن مدل رگرسیون خود بجای سال و مقدار، از مقادیر بدست آمده و مقدار استفاده میکنیم (بجای ایکس از مقادیر تابع استفاده میکنیم).

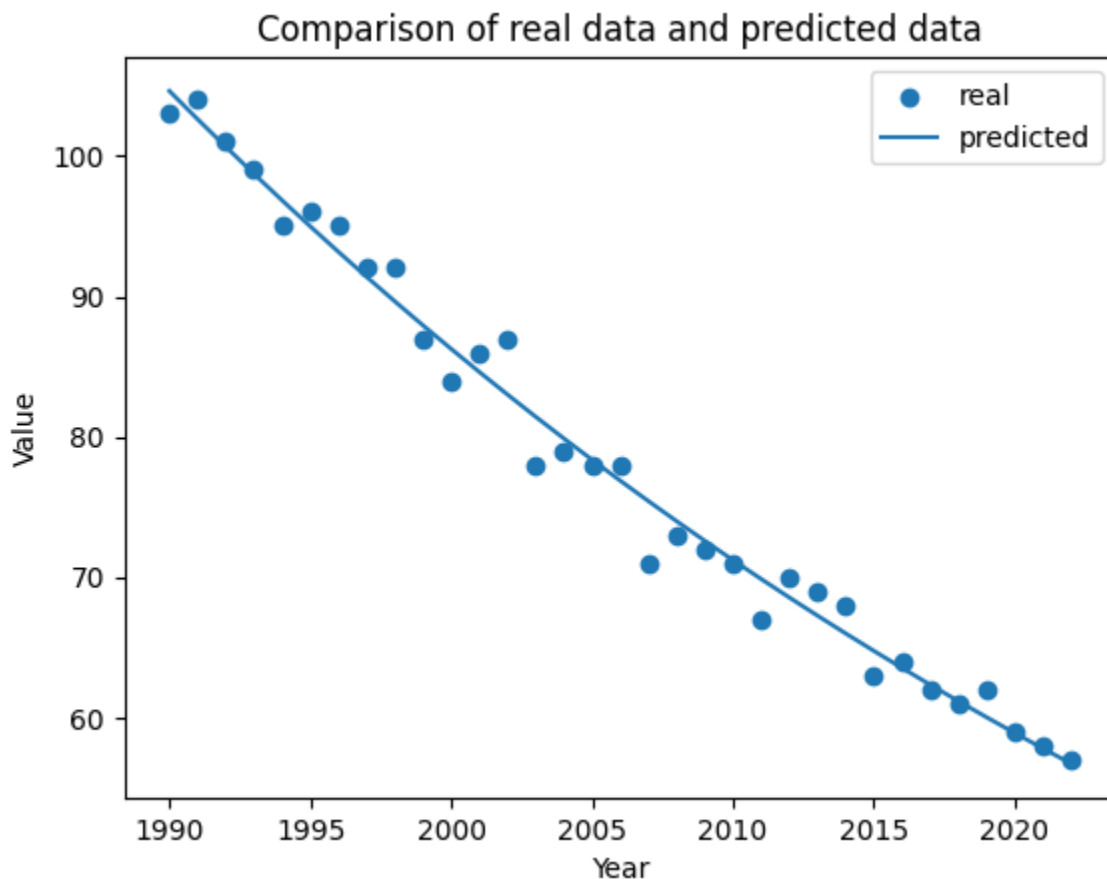
```
[array([1.]),
 array([0.98019867]),
 array([0.96078944]),
 array([0.94176453]),
 array([0.92311635]),
 array([0.90483742]),
 array([0.88692044]),
 array([0.86935824]),
 array([0.85214379]),
 array([0.83527021]),
 array([0.81873075]),
 array([0.8025188]),
 array([0.78662786]),
 array([0.77105159]),
 array([0.75578374]),
 array([0.74081822]),
 array([0.72614904]),
 array([0.71177032]),
 array([0.69767633]),
 array([0.68386141]),
 array([0.67032005]),
 array([0.65704682]),
 array([0.64403642]),
 array([0.63128365]),
 array([0.61878339]),
 ...,
 array([0.32627979]),
 array([0.31981902]),
 array([0.31348618]),
 array([0.30727874]),
 array([0.30119421])]
```

8) سپس مدل خود را فیت میکنیم (طبق گفته های سوال مقادیر ایکس را مقادیر بدست آمده بالا و همچنین مقادیر وای را مقادیر تقاضا میگذاریم)

```
X = np.array(demand).reshape(-1,1)
Y = df['demand'].values.reshape(-1,1)
model.fit(X,Y)
```

```
▼ LinearRegression
LinearRegression()
```

9) سپس طبق اطلاعات بدست آمده از مدل ، برای هر سال ابتدا مقادیر را پیشبینی میکنیم ، سپس هر دو را بر روی نمودار نشان میدهیم و آن ها را مقایسه میکنیم.



```
from sklearn.metrics import r2_score

r2 = r2_score(value, y)
print("R-squared: ", r2)

✓ 0.0s

R-squared: 0.9849573838494837
```

مشاهده میکنیم که شاخص آر توی مدل بدست آمده بالا است .

10) رگرسیون خطی قیمت و سال را بدست می آوریم.

```
regresion(df['Years'],df['price'])
```

```
(-248528.5, 125.5)
```

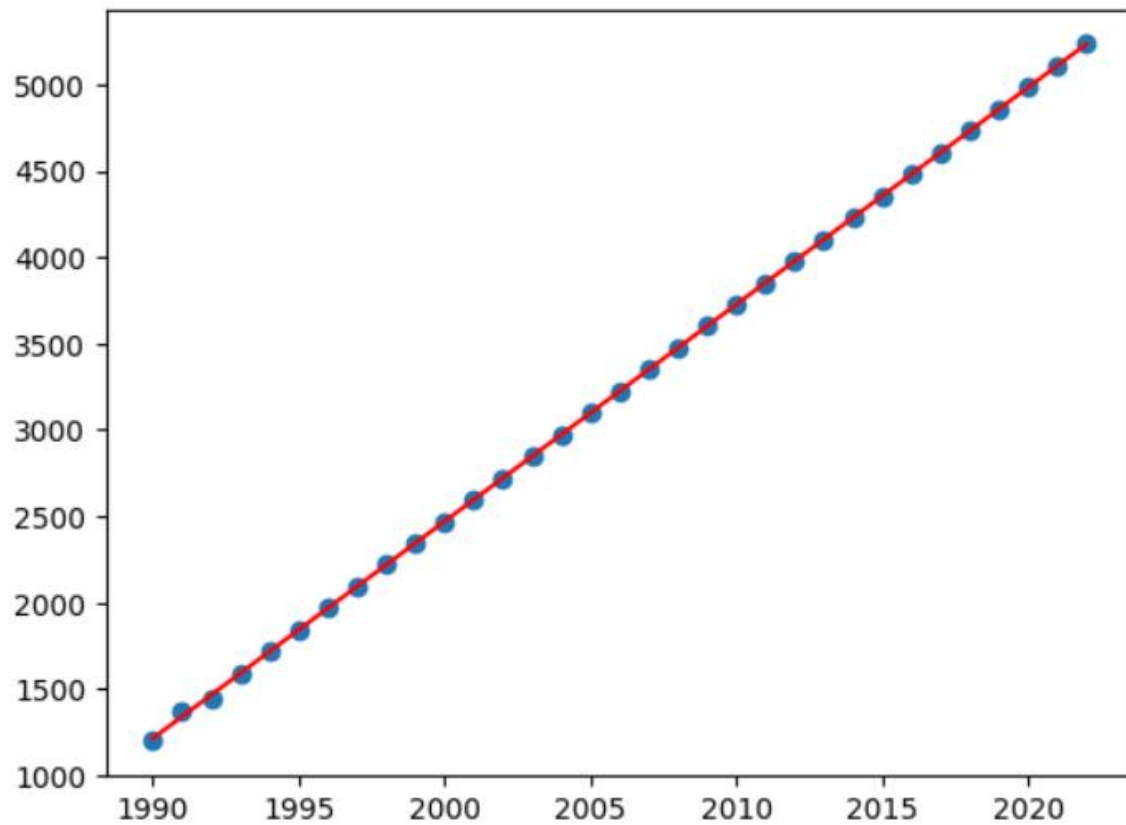
مقادیر قیمت را تا سال 2050 را پیش بینی می کنیم.

```
predicted_list=[]  
for i in range(2023,2051):  
    predicted_list.append((i,predict(i,df['Years'],df['price'])))  
predicted_list
```

```
[(2023, 5358.0),  
(2024, 5483.5),  
(2025, 5609.0),  
(2026, 5734.5),  
(2027, 5860.0),  
(2028, 5985.5),  
(2029, 6111.0),  
(2030, 6236.5),  
(2031, 6362.0),  
(2032, 6487.5),  
(2033, 6613.0),  
(2034, 6738.5),  
(2035, 6864.0),  
(2036, 6989.5),  
(2037, 7115.0),  
(2038, 7240.5),  
(2039, 7366.0),  
(2040, 7491.5),  
(2041, 7617.0),  
(2042, 7742.5),  
(2043, 7868.0),  
(2044, 7993.5),  
(2045, 8119.0),  
(2046, 8244.5),  
(2047, 8370.0),  
(2048, 8495.5),  
(2049, 8621.0),  
(2050, 8746.5)]
```

11) تابع نقطه ای و خطی را رسم کردیم و با هم مقایسه می کنیم.

```
pltreg(df['Years'],df['price'])
```



دقت رگرسیون را بدست می آوریم.

```
score(df['Years'],df['price'])
```

0.9999653378861689

با توجه به دقت رگرسیون بدست آمده که تقریباً مساوی 100٪ است نشان دهنده این است که تغییرات قیمتی بر اساس یک ضریب مشخص بالا و پایین می شود و تاثیرات تورمی بر روی آن زیاد نیست.

خواسته ها:

1) ابتدا برای هر ستون ، یک مدل رگرسیون فیت میکنیم، سپس داده های هر ستون را پیشبینی کرده و در ستون مد نظر ذخیره میکنیم، در مرحله بعد تخفیفات گفته شده را اعمال میکنیم و در انتها سود هر سال را طبق داده های پیشبینی شده ، بدست میاوریم.

```
2023 91202.87188038634
2024 87175.28851841885
2025 82882.2802762446
2026 78323.84715385642
2027 73499.98915125956
2028 68410.70626845071
2029 63055.998505434225
2030 57435.865862206265
2031 51550.308338766365
2032 44885.269381457125
2033 38480.20083882823
2034 31810.328625761234
2035 24875.65274225312
2036 17676.173188309156
2037 10211.889963922295
2038 2482.803069096204
Year : 2039 profit : -5511.08749616993 *****
```

مشاهده میکنیم که تا سال 2039 میتوانیم سود بکنیم.

2) با استفاده از سود پیشبینی شده در قسمت قبل و همچنین فاکتور پی به اف که قبلا تعریف شده است ، تمامی داده ها را به سال صفر میاوریم سپس با استفاده از فاکتور ای به پی ، پرداخت های سالانه خواسته شده را بدست میاوریم.

```
p_value=0
for i in range(9):
    p+=fp_formula(future_profit[i]*0.05, i+1)
    #print(future_profit[i])
print(f'present value : {p}')
A = ((1 - (1 + 0.05)**(-10)) / i)*p
print(f'A : {A}')
✓ 0.0s

present value : 26576.006921931457
A : 1282.5805057958478
```

3) در این خواسته از ما خواسته شده است که یک خریدار اگر بخواهد این شرکت را خریداری کند با چه قیمتی بخرد که حداقل 250000 دلار در سال پایه به او سود بدهد.

ابتدا ما تابعی تعریف کردیم که ارزش های آینده را به سال ابتدایی برگرداند.

```
def FOM1(i,n,F):  
    P = F / (1 + i) ** n  
    return P
```

در ادامه هزینه ها و درآمد های این سال ها را محاسبه کرده ایم.

```
a=0  
for j in range(0,33):  
    A = [j,0]  
    x = df.iloc[j,0]  
    i = 0.05  
    n = x - 1990  
    sal = df.iloc[j,1]  
    mat1 = df.iloc[j,2]  
    mat2 = df.iloc[j,3]  
    dem = df.iloc[j,4]  
    price = df.iloc[j,5]  
    if dem > 60:  
        F = -sal + (price - mat1 - (mat2 - 0.18 * mat2)) * dem  
        a += FOM1(i,n,F)  
    elif dem <= 60 and dem > 40:  
        F = -sal + (price - mat1 - (mat2 - 0.1 * mat2)) * dem  
        a += FOM1(i,n,F)  
    elif dem <= 40 and dem > 20:  
        F = -sal + (price - mat1 - (mat2 - 0.05 * mat2)) * dem  
        a += FOM1(i,n,F)  
    else :  
        F = -sal + (price - mat1 - mat2) * dem  
        a += FOM1(i,n,F)  
  
a
```

هزینه های ما در ابتدا حقوقی است که باید به کارمندان شرکت در طی این سال ها می دادیم و هزینه دیگر هزینه خرید متریال یک و متریال دو است که باید از درآمد ما این مقادیر کم شوند.

هزینه متریال ها و قیمت ها بستگی به تعداد همان سال بوده است.

با توجه به خواسته اول ما به ازای خرید مقادیر مختلف متریکال دوم تخفیفات متفاوتی دریافت می‌کنیم.

تابع شرطی تعریف کردیم که اگر از 60 عدد تقاضا بیشتر بود 18٪ تخفیف اگر بین 40 تا 60 عدد تقاضا 10٪ تخفیف و بین 20 تا 40 عدد تقاضا 5٪ تخفیف برای متریکال دوم وجود داشته باشد.

در نهایت قیمت را از هزینه های متریکال ها کم و در تعداد تقاضا ضرب می‌کنیم و حقوق کارمندان را نیز کم کرده و عدد بدست آمده را به سال اول (1990) می‌بریم. (نرخ بهره داده نشده است با توجه به طرح تمرین که 5 درصد بود 5 درصد در نظر گرفتیم).

در نهایت ارزش بدست آمده در سال 1990 به شرح زیر است:

a

1725352.3214124835

حال ما می‌خواهیم به ارزش 250000 در همان سال سود داشته باشیم پس خواهیم داشت:

```
p = a + 250000
i = 0.05
n = 2022 - 1990
POM1(i,n,p)
```

9412438.191400751

با استفاده از تابعی که در بالا تعریف کرده بودیم (POM1) ارزش تمامی مقادیر بعلاوه حداقل سودمان را به ارزش سال 2022 تبدیل می‌کنیم.