



تمرین پایتون درس اقتصاد مهندسی

استاد درس:

دکتر رضاپور نیاری

گزارش کار

نام و نام خانوادگی	شماره دانشجویی
صبا عبدی	۴۰۱۱۰۴۲۷۶
پردیس رحیمی نصر	۴۰۱۱۷۰۴۰۱
بیتا باقر	۴۰۱۱۰۳۹۱۱
نیما باقرنژاد	۴۰۱۱۰۳۹۲۲

فهرست مطالب:

۲.....	مقدمه:
۳.....	بخش اول:
۴.....	بخش دوم:
۵.....	بخش سوم:
۶.....	بخش چهارم:
۷.....	بخش پنجم:
۸.....	بخش ششم:
۹.....	بخش هفتم:
۱۰.....	بخش هشتم:
۱۰.....	بخش نهم:
۱۱.....	بخش دهم:
۱۲.....	بخش یازدهم:
۱۳.....	خواسته‌ها:
۱۳.....	خواسته‌ی اول:
۱۴.....	خواسته دوم:
۱۴.....	خواسته سوم:
۱۵.....	چالش‌ها:

مقدمه:

در ابتدا، برای انجام این تمرین، تقسیم‌بندی انجام شد و قرار شد هر کدام از اعضای گروه انجام بخشی از تمرین را بر عهده بگیرند. به این صورت که یک نفر مسئولیت نوشتن توابع اقتصاد مهندسی در پایتون را عهده‌دار شود و سه نفر دیگر به انتخاب خود هر کدام روی بخشی از فایل ژوپیتر کار کنند و در نهایت یک نفر مسئولیت نوشتن heading ها و mark down ها و یک نفر نیز مسئولیت نوشتن گزارش کار را بر عهده بگیرد.

در بخش اول یک ریپازیتوری بر روی گیت ایجاد شد که همه‌ی اعضای گروه بتوانند از طریق آن با یکدیگر در ارتباط باشند. سپس در فایل ژوپیتر تعدادی از کتابخانه‌های مورد نیاز و همچنین توابع اقتصاد مهندسی نوشته شده در فایل پایتون صدا زده شدند.

سپس با توجه به تمرین که نیازمند خوانش تعدادی دیتا از یک فایل اکسل بود، فایل اکسل فراخوانی شد که به سبب آن، دیتاها وارد فایل ژوپیتر شدند که در زیر قسمتی از آن قابل مشاهده است.

	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
9	1999	57189.0	337.895792	120.150370	87	2346.0
10	2000	63218.0	358.169539	123.958881	84	2471.5
11	2001	55794.0	379.659712	127.847867	86	2597.0
12	2002	56957.0	402.439294	131.815854	87	2722.5
13	2003	NaN	426.585652	135.861021	78	2848.0
14	2004	57081.0	452.180791	139.981164	79	2973.5
15	2005	61111.0	479.311639	144.173669	78	3099.0
16	2006	67938.0	508.070337	148.435472	78	3224.5
17	2007	66029.0	538.554557	152.763021	71	3350.0
18	2008	83088.0	570.867831	157.152236	73	3475.5
19	2009	81363.0	605.119900	161.598465	72	3601.0

عکس شماره ۱

بعد از فراخوانی فایل اکسل، به بررسی و حل بخش‌های مختلف تمرین می‌پردازیم.

بخش اول:

بخش اول از ما خواسته است تا بخش‌های خالی جدول را به روش و شیوه‌ی درست پر کنیم. برای این کار نیاز داریم تا یک دید کلی نسبت به دیتای خود داشته باشیم و برای این کار از تابع `info()` استفاده می‌کنیم و نتیجه‌ی زیر را می‌گیریم:

```
company_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33 entries, 0 to 32
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype  
---  -
0   Years       33 non-null    int64   
1   Salary      30 non-null    float64  
2   material 1  33 non-null    float64  
3   material 2  33 non-null    float64  
4   demand      33 non-null    int64   
5   price       33 non-null    float64  
dtypes: float64(4), int64(2)
memory usage: 1.7 KB
```

عکس شماره ۲

طبق خروجی داده شده، این دیتا شامل ۳ قسمت خالی در ستون `salary` می‌باشد که نیازمند پر شدن است.

برای پر کردن این بخش‌ها، ابتدا از میانگین استفاده می‌کنیم که بعد از محاسبه‌ی آن و چک کردن دوباره‌ی ستون `salary` به این موضوع می‌رسیم که روند این ستون به صورت صعودی است و بنابراین استفاده از میانگین برای پر کردن بخش‌های خالی درست نمی‌باشد. پس، از متود `bfill` استفاده کردیم که برای هر قسمت خالی، داده و مقدار قسمت بعدی آن را می‌نویسد. خروجی را در زیر می‌توانید مشاهده کنید.

	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
9	1999	57189.0	337.895792	120.150370	87	2346.0
10	2000	63218.0	358.169539	123.958881	84	2471.5
11	2001	55794.0	379.659712	127.847867	86	2597.0
12	2002	56857.0	402.439294	131.815854	87	2722.5
13	2003	57081.0	426.585652	135.861021	78	2848.0
14	2004	57081.0	452.180791	139.981164	79	2973.5
15	2005	61111.0	479.311639	144.173669	78	3099.0
16	2006	67938.0	508.070337	148.435472	78	3224.5
17	2007	66029.0	538.554557	152.763021	71	3350.0
18	2008	83088.0	570.867831	157.152236	73	3475.5
19	2009	81363.0	605.119900	161.598465	72	3601.0
20	2010	93940.0	641.427094	166.096436	71	3726.5

عکس شماره ۳

اگر دقت کنید، در عکس شماره‌ی ۱ در ردیف ۱۳ مقداری برای ستون `salary` وجود ندارد ولی در عکس شماره ۳ و بعد از استفاده از متود `bfill` همانطور که مشاهده می‌کنیم، این بخش، مقدار ردیف ۱۴ را برای خود گرفته است.

بخش دوم:

در این بخش از ما خواسته شده است تا با استفاده از رگرسیون خطی، حقوق چند سال آینده‌ی کارکنان را پیشبینی کنیم. بنابراین با استفاده از کتابخانه‌ی sklearn رگرسیون را فراخوانی می‌کنیم. سپس نیاز داریم تا مدل را بر روی دو متغیر فیت کنیم. بعد از انجام این کار، زمان نوشتن تابع مورد نیاز می‌باشد و با استفاده از آن، پیشبینی حقوق برای سال‌های ۲۰۲۳ الی ۲۰۵۰ صورت می‌گیرد که بخشی از آن در عکس شماره ۴ آمده است.

	Years	Salary
0	2023	122055.715909
1	2024	124726.758021
2	2025	127397.800134
3	2026	130068.842246
4	2027	132739.884358
5	2028	135410.926471
6	2029	138081.968583
7	2030	140753.010695
8	2031	143424.052807
9	2032	146095.094920
10	2033	148766.137032
11	2034	151437.179144
12	2035	154108.221257
13	2036	156779.263369
14	2037	159450.305481
15	2038	162121.347594
16	2039	164792.389706
17	2040	167463.431818
18	2041	170134.473930
19	2042	172805.516043
20	2043	175476.558155
21	2044	178147.600267
22	2045	180818.642380
23	2046	183489.684492
24	2047	186160.726604

عکس شماره ۴

حال، نیاز است تا همه‌ی حقوق‌ها از سال ۱۹۹۰ الی ۲۰۵۰ به ترتیب در یک جدول قرار بگیرند که نتیجه را می‌توان در عکس شماره ۵ مشاهده کرد.

```
all_salary = pd.concat([previous_salary, predicted_salary], ignore_index=True)
all_salary
```

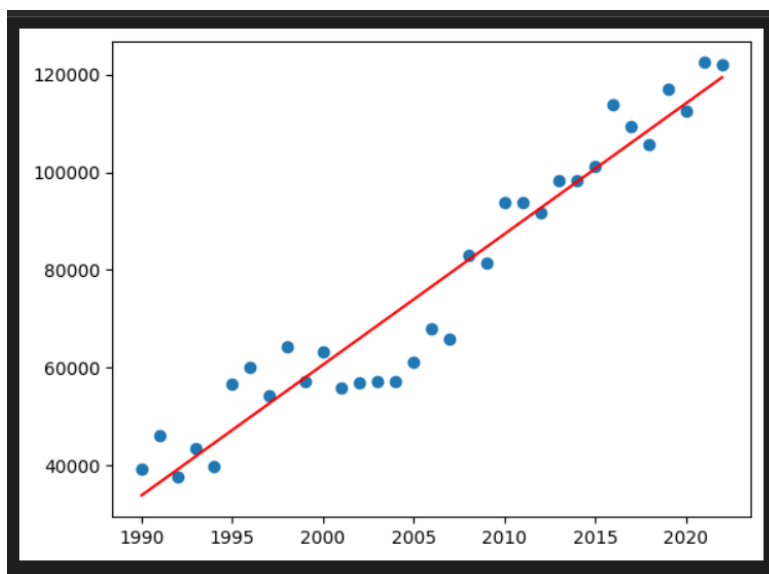
	Years	Salary
0	1990	39343.000000
1	1991	46205.000000
2	1992	37731.000000
3	1993	43525.000000
4	1994	39891.000000
...
56	2046	183489.684492
57	2047	186160.726604
58	2048	188831.768717
59	2049	191502.810829
60	2050	194173.852941

61 rows × 2 columns

عکس شماره ۵

بخش سوم:

در این بخش ابتدا پیشبینی مدل در بخش قبلی را فراخوانی کرده و سپس با استفاده از کتابخانه `matplotlib` نمودار خطی رگرسیون و نمودار نقطه‌ای دیتاهای موجود را رسم می‌کنیم که در زیر می‌توان آن را مشاهده کرد.



عکس شماره ۶

سپس، میزان دقت مدل را با استفاده از متود `score()` بررسی می‌کنیم که باتوجه به خروجی، این میزان ۹۲ درصد می‌باشد که مقدار قابل قبولی است.

```
score = model.score(X_salary, Y_salary)
score
0.9298401700878312
```

عکس شماره ۷

بخش چهارم:

در این بخش، با توجه به راهنمایی خود تمرین، ابتدا نرخ افزایش قیمت محصول را برابر با جمع نرخ بهره و نرخ تورم قرار می‌دهیم که خروجی آن عدد زیر خواهد شد.

```
rate = 0.05
inflation = 0.01
material1_rate = rate + inflation
material1_rate
```

0.060000000000000005

عکس شماره ۸

سپس با استفاده از تابع f/p اقتصاد مهندسی، قیمت محصول ۱ در سال‌های بعدی را به دست می‌آوریم و در نهایت با کنار هم گذاشتن قیمت محصول ۱ در سال‌های ۱۹۹۰ الی ۲۰۵۰ به روند زیر دست پیدا می‌کنیم.

	Years	material 1
0	1990	200.000000
1	1991	212.000000
2	1992	224.720000
3	1993	238.203200
4	1994	252.495392
...
56	2046	1565.647845
57	2047	1569.336533
58	2048	1572.894196
59	2049	1576.330091
60	2050	1579.652518

61 rows × 2 columns

عکس شماره ۹

بخش پنجم:

در این بخش، ابتدا قیمت محصول ۲ در سال‌های ۱۹۹۰ الی ۲۰۲۲ را در سال پایه که ۱۹۹۰ می‌باشد، حساب می‌کنیم که نتیجه به شرح زیر می‌باشد:

	Years	material 2
0	1990	89.60
1	1991	88.25
2	1992	86.90
3	1993	85.55
4	1994	84.20
5	1995	82.85
6	1996	81.50
7	1997	80.15
8	1998	78.80
9	1999	77.45
10	2000	76.10
11	2001	74.75
12	2002	73.40
13	2003	72.05
14	2004	70.70
15	2005	69.35
16	2006	68.00
17	2007	66.65
18	2008	65.30
19	2009	63.95
20	2010	62.60
21	2011	61.25
22	2012	59.90

عکس شماره ۱۰

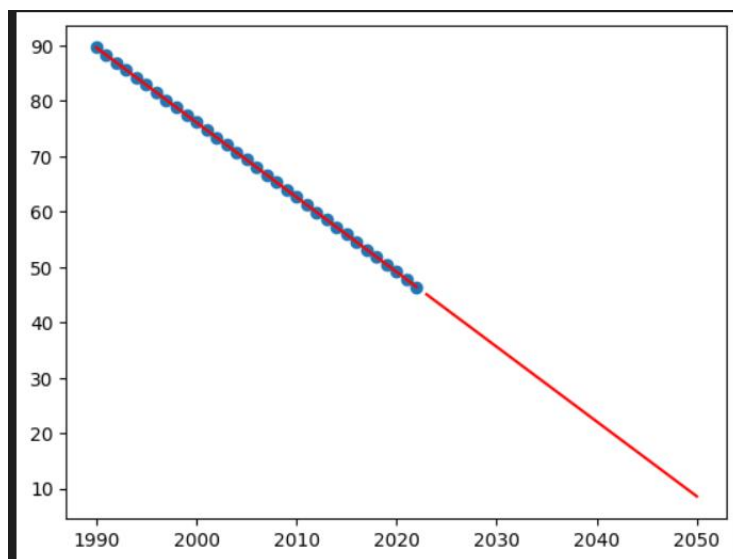
سپس باتوجه به اینکه تقاضا برای محصول ۲ هر سال کاهش یافته است، با استفاده از رگرسیون و تابع نوشته شده، قیمت واقعی محصول در سال‌های آینده را پیدا می‌کنیم اما باتوجه به اینکه در قسمت قبلی این قیمت‌ها در سال پایه حساب شده بودند و تابع به صورتی نوشته شده است که قیمت‌ها را در سال پایه حساب می‌کند، بنابراین با استفاده از تابع حال به آینده‌ی اقتصاد مهندسی، این قیمت‌ها را در آینده هم محاسبه می‌کنیم که بخشی از نتیجه به صورت زیر می‌شود.

	Years	material 2
0	2023	225.393644
1	2024	229.571306
2	2025	233.603251
3	2026	237.464462
4	2027	241.127785
5	2028	244.563780
6	2029	247.740555
7	2030	250.623598
8	2031	253.175594
9	2032	255.356231
10	2033	257.121992

عکس شماره ۱۱

بخش ششم:

در این بخش با استفاده از کتابخانه `matplotlib` نمودار خطی رگرسیون و نمودار نقطه‌ای دیتاهای موجود مورد نیاز را رسم می‌کنیم.



عکس شماره ۱۲

سپس با استفاده از متود `score()` مقدار دقت مدل را به دست می‌آوریم که با توجه به خروجی بیش از ۹۰ درصد می‌باشد.

```
score = model_mterial2.score(X_material2, Y_material2)
score
1.0
```

عکس شماره ۱۳

بخش هفتم:

در این بخش، ابتدا سال‌های ۲۰۲۳ الی ۲۰۵۰ را به دیتافریم اصلی اضافه می‌کنیم که نمونه‌ی آن را می‌توان در عکس شماره ۱۴ مشاهده کرد.

	Years	Salary	material 1	material 2	demand	price
0	1990	39343.0	200.000000	89.600000	103.0	1200.0
1	1991	46205.0	212.000000	92.662500	104.0	1375.0
2	1992	37731.0	224.720000	95.807250	101.0	1451.0
3	1993	43525.0	238.203200	99.034819	99.0	1593.0
4	1994	39891.0	252.495392	102.345626	95.0	1718.5
...
56	2046	NaN	NaN	NaN	NaN	NaN
57	2047	NaN	NaN	NaN	NaN	NaN
58	2048	NaN	NaN	NaN	NaN	NaN
59	2049	NaN	NaN	NaN	NaN	NaN
60	2050	NaN	NaN	NaN	NaN	NaN

61 rows × 6 columns

عکس شماره ۱۴

سپس فرمول داده شده در فایل تمرین را به صورت تابع نوشته و برای سال‌های ۱۹۹۰ الی ۲۰۵۰ حساب کرده و ستون جدیدی به نام demand_function ایجاد کرده و این مقادیر را وارد آن می‌کنیم.

	Years	Salary	material 1	material 2	demand	price	demand_function
0	1990	39343.0	200.000000	89.600000	103.0	1200.0	1.000000
1	1991	46205.0	212.000000	92.662500	104.0	1375.0	0.980199
2	1992	37731.0	224.720000	95.807250	101.0	1451.0	0.960789
3	1993	43525.0	238.203200	99.034819	99.0	1593.0	0.941765
4	1994	39891.0	252.495392	102.345626	95.0	1718.5	0.923116
...
56	2046	NaN	NaN	NaN	NaN	NaN	0.326280
57	2047	NaN	NaN	NaN	NaN	NaN	0.319819
58	2048	NaN	NaN	NaN	NaN	NaN	0.313486
59	2049	NaN	NaN	NaN	NaN	NaN	0.307279
60	2050	NaN	NaN	NaN	NaN	NaN	0.301194

61 rows × 7 columns

عکس شماره ۱۵

بخش هشتم:

برای پیشبینی کردن تقاضا در سال‌های آینده، مدل رگرسیونی پیاده‌سازی می‌کنیم که در آن هم ضریب تقاضا و هم تقاضای اصلی موجود می‌باشد و سپس بر اساس این مدل، تابع پیشبینی تقاضا را می‌نویسیم و از آن برای پیشبینی و پر کردن مقادیر خالی ستون demand استفاده می‌کنیم مانند شکل زیر.

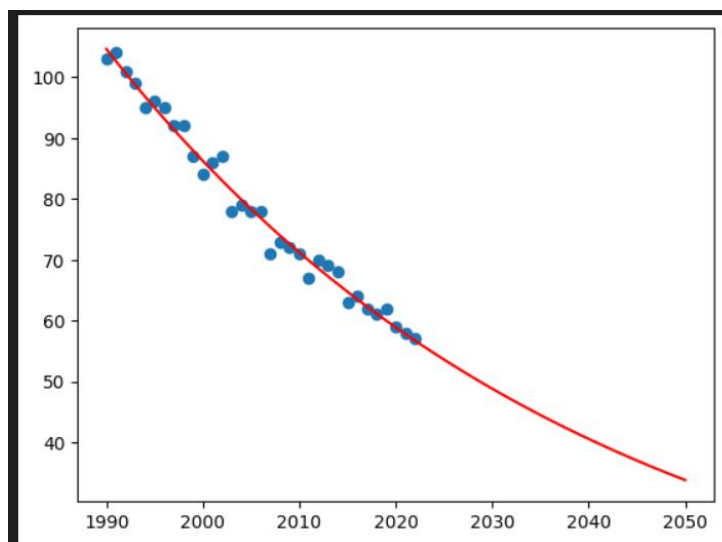
	Years	Salary	material 1	material 2	demand	price	demand_function
0	1990	39343.0	200.000000	89.600000	103.000000	1200.0	1.000000
1	1991	46205.0	212.000000	92.662500	104.000000	1375.0	0.980199
2	1992	37731.0	224.720000	95.807250	101.000000	1451.0	0.960789
3	1993	43525.0	238.203200	99.034819	99.000000	1593.0	0.941765
4	1994	39891.0	252.495392	102.345626	95.000000	1718.5	0.923116
...
56	2046	NaN	NaN	NaN	36.339007	NaN	0.326280
57	2047	NaN	NaN	NaN	35.684136	NaN	0.319819
58	2048	NaN	NaN	NaN	35.042232	NaN	0.313486
59	2049	NaN	NaN	NaN	34.413039	NaN	0.307279
60	2050	NaN	NaN	NaN	33.796305	NaN	0.301194

61 rows x 7 columns

عکس شماره ۱۶

بخش نهم:

در این بخش نیز مانند بخش‌های پیشین، با استفاده از کتابخانه‌ی matplotlib نمودار خطی برای رگرسیون و نمودار نقطه‌ای برای دیتاهای داده شده رسم می‌کنیم که خروجی در زیر آمده است:



عکس شماره ۱۷

سپس دقت مدل را به دست می‌آوریم که با توجه به عکس شماره ۱۸ این دقت برابر با ۹۸ درصد می‌باشد.

```
demand_score = model_demand.score(X_demand_factor, Y_demand)
demand_score
```

0.9849573838479752

عکس شماره ۱۸

بخش دهم:

در این بخش، با استفاده از دیتاهای در دسترس و ایجاد مدل مناسب و استفاده از آن برای نوشتن تابع پیشبینی، دیتا فریم جدیدی ایجاد کرده و مقادیر به دست آمده برای سال‌های آینده را در آن وارد می‌کنیم.

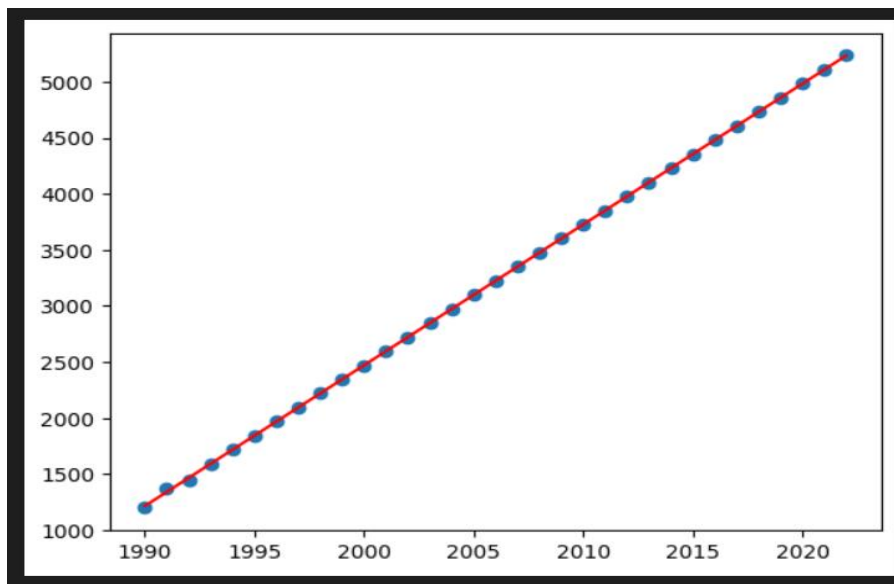
بخشی از این جدول به شکل زیر می‌باشد:

	Years	price
0	2023	5358.0
1	2024	5483.5
2	2025	5609.0
3	2026	5734.5
4	2027	5860.0
5	2028	5985.5
6	2029	6111.0
7	2030	6236.5
8	2031	6362.0
9	2032	6487.5
10	2033	6613.0
11	2034	6738.5
12	2035	6864.0
13	2036	6989.5
14	2037	7115.0
15	2038	7240.5
16	2039	7366.0
17	2040	7491.5
18	2041	7617.0
19	2042	7742.5
20	2043	7868.0
21	2044	7993.5
22	2045	8119.0
23	2046	8244.5
24	2047	8370.0

عکس شماره ۱۹

بخش یازدهم:

در این بخش نیز باید با استفاده از کتابخانه‌ی `matplotlib`، نمودار نقطه‌ای دیتاها و نمودار خطی رگرسیون را رسم کنیم که خروجی در عکس شماره ۲۰ قابل مشاهده است.



عکس شماره ۲۰

سپس دقت مدل را به دست می‌آوریم که با توجه به محاسبات، دقت برابر ۹۹.۹۹ درصد می‌باشد که به شدت دقت خوبی است.

```
score_price_prediction = model_final_price.score(X_final_price, Y_final_price)
score_price_prediction
```

```
0.9999653378861689
```

عکس شماره ۲۱

خواسته‌ها:

خواسته‌ی اول:

قبل از هرکاری دیتابیس خود را با داده‌های قبلی و داده‌های جدید پیشبینی شده، پر می‌کنیم.

	Years	Salary	material 1	material 2	demand	price	demand_function
0	1990	39343.000000	200.000000	89.600000	103.000000	1200.0	1.000000
1	1991	46205.000000	212.000000	92.662500	104.000000	1375.0	0.980199
2	1992	37731.000000	224.720000	95.807250	101.000000	1451.0	0.960789
3	1993	43525.000000	238.203200	99.034819	99.000000	1593.0	0.941765
4	1994	39891.000000	252.495392	102.345626	95.000000	1718.5	0.923116
...
56	2046	183489.684492	1565.647845	260.978022	36.339007	8244.5	0.326280
57	2047	186160.726604	1569.336533	262.002218	35.684136	8370.0	0.319819
58	2048	188831.768717	1572.894196	263.026414	35.042232	8495.5	0.313486
59	2049	191502.810829	1576.330091	264.050610	34.413039	8621.0	0.307279
60	2050	194173.852941	1579.652518	265.074806	33.796305	8746.5	0.301194

61 rows × 7 columns

عکس شماره ۲۲

حالا، باتوجه به خواسته‌ی اول که در واقع محاسبه‌ی سود می‌باشد، نیاز به درآمدها و هزینه‌های هر سال داریم که با نوشتن توابعی برای محاسبه‌ی آن‌ها، به آن‌ها می‌رسیم.

اولین تابعی که از آن استفاده می‌کنیم، تابع `material1_cost_prediction` می‌باشد که در این تابع، هزینه‌ی محصول ۱ ما محاسبه می‌شود. در این تابع ابتدا بررسی می‌شود که آیا دیتای مورد نیاز موجود می‌باشد یا خیر و اگر موجود نبود، آن را پیشبینی می‌کند.

سپس تابع `material2_cost_prediction` نوشته شد که نحوه‌ی کارکرد آن مانند تابع قبلی بود با این فرق که در این تابع از توابع اقتصاد مهندسی استفاده نشد و صرفاً از مدل رگرسیونی که فیت کرده بودیم استفاده کردیم.

سپس توابع دیگر را مانند دو تابع بالا تعریف کردیم. توابع دیگر از جمله توابع برای محاسبه‌ی `salary`، `price`، `demand` و `profit` بودند.

بعد از تعریف این توابع، با استفاده از تابع `profit()`، سود هر سال را حساب می‌کنیم تا به سالی برسیم که سود منفی شود و می‌فهمیم از آن سال به بعد شرکت سودده نخواهد بود.

profit(2037)
2937.7787096361862
profit(2038)
-5190.880599573255

عکس شماره ۲۳

با توجه به عکس شماره ۲۳، سال ۲۰۳۷ آخرین سالی است که در آن سود مثبت می‌باشد و شرکت از سال ۲۰۳۸ به بعد ضررده می‌شود.

خواسته دوم:

با توجه به توضیحات این خواسته، نرخ بهره را ۵ درصد در نظر گرفته و سود را در سال پایه‌ی ۲۰۲۱ حساب کرده و سپس سرمایه‌گذاری سالانه را به دست می‌آوریم که میزان آن ۳۷۱۲ می‌باشد.

```
base_year = 2022
investment_money_converted_to_base_year = [f_to_p(year - base_year, 0.05, profit(year) * 0.25) for year in range(2022, 2033)]
total_investment = sum(investment_money_converted_to_base_year)*0.2
annual_investment = p_to_a(11, 0.05, total_investment)
annual_investment
```

3712.2195727508556

عکس شماره ۲۴

خواسته سوم:

در این بخش ابتدا سودها را به سال پایه برگردانده و سپس قیمت خرید مناسب را به خریدار پیشنهاد می‌دهیم که این مقدار ۱۶۴۷۲۹.۴ می‌باشد.

```
base_year = 2022
profit_converted_to_base_year = [f_to_p(year - base_year, 0.05, profit(year)) for year in range(2023, 2051)]
company_value_in_base_year = sum(profit_converted_to_base_year)
company_value_in_base_year += profit(2022)
minimum_price_for_customer = company_value_in_base_year - 250000
minimum_price_for_customer
```

164729.3914575706

عکس شماره ۲۵

چالش‌ها:

بیشترین و بزرگترین چالشی که به آن برخورد کردیم، مربوط به خواسته‌ی سوم می‌شد. به دلیل اینکه سال پایه را ۱۹۹۰ در نظر گرفته بودیم و مبلغ نهایی پیشنهادی، منفی در می‌آمد که توجیحی هم نداشت.

در نهایت به این نتیجه رسیدیم که باید سال پایه را ۲۰۲۲ که زمان خرید فرد خریدار است در نظر بگیریم تا مبلغ نهایی منطقی به دست بیاید.

همچنین در بخش پنجم هم در ابتدا به این صورت در نظر گرفته بودیم که از همان ابتدا مبلغ را در سال آینده محاسبه و پیشبینی کنیم و رگرسیون بزنیم و سپس دقت آن مدل ۹۰ درصد به دست آمد که در برابر دقت ۱۰۰ در صدی مدل کنونی بسیار کمتر می‌باشد.