

به نام خدا



K. N. Toosi University of Technology

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده برق

اقتصاد مهندسی

گزارش پروژه کد نویسی

فربد رئیسی

40005273

استاد: آقای دکتر نوذریان

تیر 1403

فهرست مطالب

شماره صفحه	عنوان
3.....	بررسی نرم افزار به طور کلی.
4.....	بررسی کد و تئوری با جزئیات
4.....	شروع برنامه
4.....	کتابخانه های استفاده شده
5.....	توابع محاسبه پارامتر های اقتصادی در ابتدای نرم افزار
6.....	تابع ارزیابی اقتصادی یا calculate_economic_metrics
9.....	تابع نمودار جریان مالی یا plot_cash_flow
11.....	توابع () start1() و start2()
13.....	حل یک مثال برای start2()
14.....	مراجع

بررسی نرم افزار به طور کلی

این نرم افزار نوشته شده برای پژوهش درس اقتصاد مهندسی است و هدف آن این است که اقتصادی بودن و یا نبودن یک پژوهش را بررسی کند، در ابتدای نرم افزار دو گزینه برای کاربر وجود دارد

کاربر میتواند با انتخاب خدمات **1**، از نرم افزاری بهره ببرد که پارامتر های Annual cost income و future worth Present worth را به یکدیگر تبدیل بکند، این بخش نرم افزار برای این طراحی شده است که کاربر اگر نیاز داشت که قبل از بررسی پژوهش اش این پارامتر ها را به یکدیگر تبدیل کند با استفاده از نرم افزار این کار را انجام دهد

حال به سراغ خدمت دوم یعنی بررسی اقتصادی بودن پژوهه می رویم، کاربر میتواند با انتخاب خدمت شماره **2** المان های پژوهه مثل ارزش فعلی و ارزش آینده و سود سالانه و نرخ بهره، نرخ مالیات، عمر پژوهه و ... را وارد کند و نرم افزار با محاسبه هایی که انجام می دهد بررسی می کند که ایا این پژوهه اقتصادی هست یا نه

این نرم افزار برای اقتصادی بودن یک پژوهه سه معیار را در نظر می گیرد

- ارزش فعلی یا NPV باید بزرگ تر از صفر باشد
- نرخ بهره IR کمتر از نرخ بازده داخلی IRR باشد
- دوره بازگشت سرمایه از عمر پژوهه کمتر باشد یا $n > \text{Payback Period}$

اگر این سه معیار برآورده شدند نتیجه گیری می شود که پژوهه اقتصادی است

بررسی کد و تئوری به همراه جزئیات

شروع برنامه

#start

برنامه ابتدا از شما تشکر می کند^{۶۳}

سپس از شما می پرسد که کدام یک از خدکات مدنظر شما است و شما باید با دادن عدد یک یا دو
انتخاب کنید بعد از آن برنامه توابع مربوطه را با استفاده از دستور شرطی صدا میزند

کتابخانه های استفاده شده

```
import numpy as np
import numpy_financial as npf
import matplotlib.pyplot as plt
```

کتابخانه NumPy برای محاسبه های عددی و کار با آرایه ها، matplotlib برای کشیدن نمودار و از یک
کتابخانه جزئی به نام NumPy_financial استفاده میکنیم

کتابخانه آخری که با جست و جو به نتیجه استفاده کردن از آن رسیدم به ما کمک میکند یک سری از
توابع اقتصادی را خیلی سریع تر و راحت تر بررسی کنیم که در بخش توضیحات بخش ها به آن اشاره

خواهم کرد

توابع محاسبه پارامتر های اقتصادی در ابتدای نرم افزار

```
def F_from_P(present_value, interest_rate, years):
    future_value = present_value * (1 + interest_rate) ** years
    return future_value

def F_from_A(annual_cash_flow, interest_rate, years):
    future_value = annual_cash_flow * ((1 + interest_rate)**years - 1) / interest_rate
    return future_value

def A_from_P(present_worth, interest_rate, years):
    eacf = (interest_rate * present_worth) / (1 - (1 + interest_rate)**-years)
    return eacf

def P_from_A(equivalent_annual_cash_flow, interest_rate, years):
    pw = equivalent_annual_cash_flow * ((1 - (1 + interest_rate)**-years) / interest_rate)
    return pw

def P_from_F(future_value, interest_rate, years):
    pw = future_value / ((1 + interest_rate) ** years)
    return pw

def A_from_F(future_value, interest_rate, years):
    eacf = future_value * interest_rate / ((1 + interest_rate)**years - 1)
    return eacf

def depreciation(present_worth, SV, n):
    depreciation = (present_worth - SV) / n;
    return depreciation
```

همانطور که در شکل بالا مشاهده می کنید، در ابتدای نرم افزار توابعی نوشته شده است که با گرفتن مقادیر مختلف، پارامتر خواسته شده را محاسبه می کد، علت استفاده از تابع این است که ما بارها ممکن است نیاز به استفاده از این کد ها داشته باشیم و تابع مانند بلوکی است که استفاده چند باره را برای ما راحت تر می کند

برای مثال در تابع اول به نام $F_{\text{from_P}}$ این تابع present worth و interest rate و years را از کاربر می گیرد و طبق فرمول

$$F = P(1 + i)^n$$

را محاسبه کرده و آن را بر میگرداند

بقيه توابع نيز به همين شكل هستند و با استفاده از رابطه رياضي موجود مقدار دلخواه را به ما مي دهند

تابع آخری تابع Depreciation نام دارد که p , n و ارزش اسقاطی را از کاربر میگیرد و به روش خط مستقیم استهلاک را حساب کرده و به کاربر برミ گرداند



همانطور که ملاحظه می شود این تابع پارامتر های مذکور در بالا را دریافت کرده و یک متغیر دیکشنری با مقادیر IRR، net_present_worth، npv > 0 and irr > ir and payback_period < n می دهد که نشان دهنده این است که آیا این پروژه اقتصادی است یا خیر

نحوه محاسبه این متغیر and گرفتن از سه شرط ذکر شده در بخش بررسی کلی است.

`is_economical = npv > 0 and irr > ir and payback_period < n`

برای محاسبه CF یا cash flows یا جریان نقدی

$$\text{Cash Flows} = \begin{cases} -C_0 & \text{in year 0} \\ C_{\text{net}} & \text{from year 1 to year } n-1 \\ C_{\text{net}} + S + \frac{F}{(1+r)^n} & \text{in year } n \end{cases}$$

همانطور که مشاهده می کنید جریان نقدی را میتوان از جمع المان های بالا بدست آورد که C_{net} از محاسبه جمع درآمد سالانه و مقدار سیو مالیاتی بدست می آید

$$\text{سالانه}_A * (\text{نرخ مالیات}-1) = A$$

$$\text{سیو مالیاتی سالانه} = \text{استهلاک} * \text{نرخ مالیات}$$

$$\text{مقدار جریان نقدی سالانه} = \text{نرخ استهلاک خط مستقیم} + A$$

جریان نقدی کلی در خط آخر کد محاسبه شده است

$$\text{net_annual_income} = ai * (1 - tr)$$

$$\text{annual_tax_saving} = dep_rate * tr$$

$$\text{net_annual_cash_flow} = \text{net_annual_income} + \text{annual_tax_saving}$$

$$\text{cash_flows} = [-pw] + [\text{net_annual_cash_flow}] * (n - 1) + [\text{net_annual_cash_flow} + sv + P_{\text{from_F}}(fw, ir, n)]$$

حال نوبت به محاسبه NPV و IRR میرسد

اینجا ما به راحتی از کتابخوانه numpy_financial بهره میگیریم و با دو دستور زیر آنها را محاسبه می کنیم

$$npv = npf.npv(ir, cash_flows)$$

$$irr = npf.irr(cash_flows)$$

جريان نقدی تجمعی

```
cumulative_cash_flows = np.cumsum(cash_flows)
```

مجموع تجمعی عناصر یک آرایه را محاسبه می کند. به عبارت دیگر، جريان نقدی تجمعی (Cumulative Cash Flow) در هر سال برابر با مجموع جريان های نقدی سال های گذشته و سال جاری است.

: اين متغير يك آرایه است که شامل جريان های نقدی تجمعی پروژه در

هر سال می شود.

تعیین دوره بازگشت سرمایه

```
payback_period = np.argmax(cumulative_cash_flows >= 0)
```

: اين عبارت يك آرایه بولی ایجاد می کند که نشان می دهد جريان نقدی تجمعی در هر سال آیا مثبت یا صفر است یا خیر.

: اين تابع اولین ایندکس آرایه ای که مقدار آن

باشد را بر می گرداند. به عبارت دیگر، اولین سالی که جريان نقدی تجمعی به صفر یا مقدار مثبت می رسد را پیدا می کند.

در نهایت مقادیری که در ابتدا ذکر شد توط تابع برگردانده می شوند

یکی از متغیر ها که به صورت دیکشنری است به صورت زیر برگردانده می شود

```
metrics = {
    'NPV': npv,
    'IRR': irr,
    'Payback Period': payback_period,
    'Is Economical': is_economical
}
```

تابع نمودار جریان مالی یا plot_cash_flow

def plot_cash_flows(cash_flows, cumulative_cash_flows, n):

- cash_flows : لیست یا آرایه‌ای از جریان‌های نقدی سالانه پروژه.
- cumulative_cash_flows : لیست یا آرایه‌ای از جریان‌های نقدی تجمعی پروژه.
- n : تعداد سال‌هایی که جریان‌های نقدی پروژه در نظر گرفته می‌شود.

توضیح تابع plot_cash_flows

تابع plot_cash_flows به منظور نمایش تصویری جریان‌های نقدی سالانه و تجمعی یک پروژه در طول یک دوره زمانی مشخص، از نمودارهای میله‌ای و خطی استفاده می‌کند. بباید هر قسمت از این تابع را به صورت جزئی توضیح دهیم:

- برای ایجاد آرایه‌ای از سال‌ها از دستور زیر استفاده می‌کنیم

```
years = np.arange(0, n + 1)
```

- دستور np.arange(0, n + 1) آرایه‌ای از سال‌ها از 0 تا n ایجاد می‌کند. این آرایه برای محور x در نمودارها استفاده خواهد شد.

- برای تنظیمات اولیه نمودار از دستور plt.figure(figsize=(12, 6)) استفاده می‌کنیم که یک شکل جدید با اندازه 12 اینچ در عرض و 6 اینچ در ارتفاع ایجاد می‌کند.

- دستور plt.subplot(2, 1, 1) یک subplot ایجاد می‌کند که ردیف اول در یک شبکه 2 ردیفه و 1 ستونه را اشغال می‌کند.

- دستور plt.bar(years, cash_flows, color='blue', alpha=0.7) یک نمودار میله‌ای از جریان‌های نقدی سالانه رسم می‌کند. میله‌ها به رنگ آبی با شفافیت 0.7 هستند.

- دستور plt.xlabel('Year') محور x را به عنوان "Year" برچسب‌گذاری می‌کند.

- دستور plt.ylabel('Annual Cash Flow') محور y را به عنوان "Annual Cash Flow" برچسب‌گذاری می‌کند.

- دستور plt.title('Annual Cash Flow Over Time') عنوان نمودار را به "Annual Cash Flow Over Time" تنظیم می‌کند.

رسم جریان‌های نقدی تجمعی

- دستور `plt.subplot(2, 1, 2)` یک ایجاد می‌کند که ردیف دوم در شبکه 2 ردیفه و 1 ستونه را اشغال می‌کند.

دستور `plt.plot(years, cumulative_cash_flows, color='green', marker='o')` یک نمودار خطی از جریان‌های نقدی تجمعی رسم می‌کند. خط به رنگ سبز است و نقاط داده‌ها با دایره مشخص شده‌اند.

- دستور `plt.xlabel('Year')` محور x را به عنوان "Year" برچسب‌گذاری می‌کند.
- دستور `plt.ylabel('Cumulative Cash Flow')` محور y را به عنوان "Cumulative Cash Flow" برچسب‌گذاری می‌کند.
- دستور `plt.title('Cumulative Cash Flow Over Time')` عنوان نمودار را به "Cumulative Cash Flow Over Time" تنظیم می‌کند.

تنظیم فاصله‌ها و نمایش نمودار

- دستور `plt.tight_layout()` فاصله بین زیرنمودارها را تنظیم می‌کند تا به خوبی در فضای شکل جا بیفتند.
- دستور `plt.show()` شکل را با نمودارها نمایش می‌دهد.

در نمودار نهايی:

1. پنل بالا (نمودار ميله‌اي): جریان‌های نقدی سالانه را در طول مدت پروژه نشان می‌دهد.
2. پنل پایین (نمودار خطی): جریان‌های نقدی تجمعی را نشان می‌دهد و به تحلیل تغییرات ارزش خالص پروژه در طول زمان کمک می‌کند.

با تحلیل این نمودارها، سهامداران می‌توانند درک بهتری از رفتار جریان نقدی داشته باشند و تصمیمات مالی بهتری برای پروژه اتخاذ کنند.

در انتهای یک مثال حل می‌کنیم و نمودار آن را نمایش می‌دهیم

توابع start1() و start2()

کار استارت یک این است که خدمت شماره یک یعنی تبدیل پارامتر ها را انجام دهد و این کار را با دریافت پارامتر های لازم و ارجاع به تابع های مذکور انجام می دهد

```
def Start1():
    quest = input('please choose:F_from_P(1), F_from_A(2), P_from_A(3), P_from_F(4), A_from_P(5), A_from_F(6)')
    interest_rate = int(input('how much is the annual rate?'))
    n = int(input('project Lifetime?(in years)'))

    if quest == '1':
        present_worth = int(input('what is your present worth?(P)'))
        F = F_from_P(present_worth, interest_rate, n)
        print('F is equal to ', F);

    elif quest == '2':
        annual_income = int(input('How much is your projects annual income?(A)'))
        F = F_from_A(annual_income, interest_rate, n)
        print('F is equal to ', F);

    elif quest == '3':
        annual_income = int(input('How much is your projects annual income?(A)'))
        P = P_from_A(annual_income, interest_rate, n)
        print('P is equal to ', P);

    elif quest == '4' :
        future_worth = int(input('what is your future worth?(F)'))
        P = P_from_F(future_worth, interest_rate, n)
        print('P is equal to ', P);

    elif quest == '5' :
        present_worth = int(input('what is your present worth?(P)'))
        A = A_from_P(present_worth, interest_rate, n)
        print('A is equal to ', A);

    elif quest == '6' :
        future_worth = int(input('what is your future worth?(F)'))
        A = A_from_F(future_worth, interest_rate, n)
        print('A is equal to ', A);
```

این کار به سادگی با چند دستور شرط و دریافت اطلاعات ساده شروع شده است و از توضیحات اضافه صرف نظر می شود

و اما تابع استارت 2

```
def Start2():
    present_worth = float(input('what is your present worth?(P)'))
    future_worth = float(input('what is your future worth?(F)'))
    annual_income = float(input('what is your projects annual income?(A)'))
    salvage_value = float(input('SV?'))
    interest_rate = float(input('how much is the annual rate?'))
    tax_rate = float(input('how much is the tax rate?'))
    n = int(input('project Lifetime?(in years)'))
    depreciation_rate = depreciation(present_worth, salvage_value, n) # Straight-line depreciation rate

    metrics, cash_flows, cumulative_cash_flows = calculate_economic_metrics(
        present_worth, future_worth, annual_income, salvage_value, interest_rate, tax_rate, n , depreciation_rate
    )

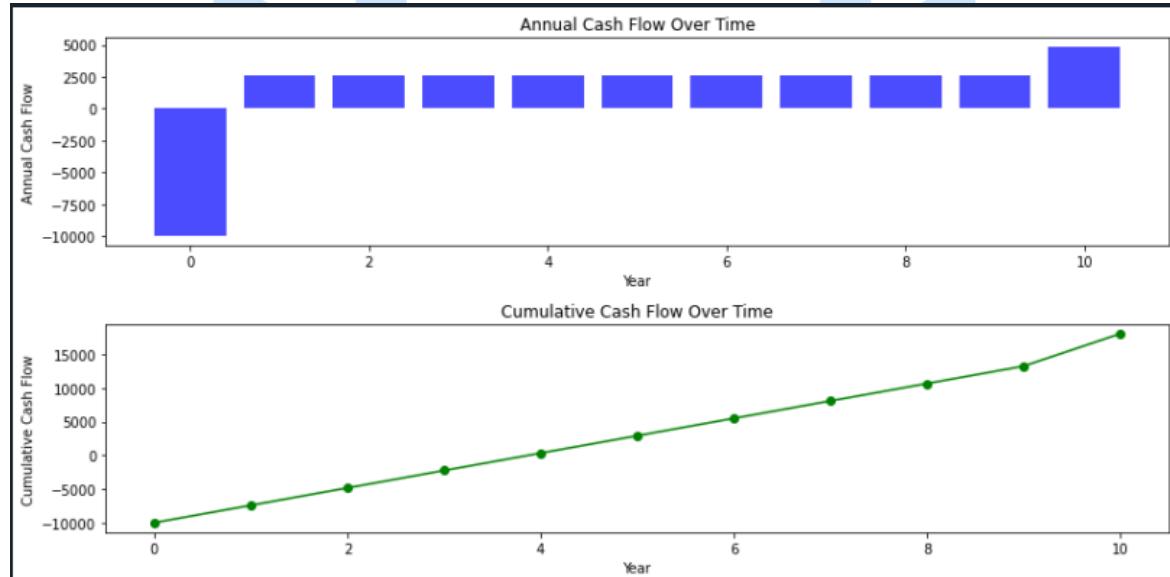
    print(f"NPV: ${metrics['NPV']:.2f}")
    print(f"IRR: {metrics['IRR']*100:.2f}%")
    print(f"Payback Period: {metrics['Payback Period']} years")
    print(f"Is the project economical? {'Yes' if metrics['Is Economical'] else 'No'}")

    plot_cash_flows(cash_flows, cumulative_cash_flows, n)
```

این تابع مقادیری که در بالا آورده شده است را از کاربر میگیرد سپس با استفاده از ارزش اسقاطی و
مقدار استهلاک خط مستقیم را محاسبه میکند و بعد مقادیر را به تابع محاسبه المان های اقتصادی که
پیش تر راجع به آن صحبت شد می دهد و مقداری گرفته شده از آن تابع را چاپ میکند و علاوه بر این با
صدا زدن تابع نمودار، نمودار را نیز رسم میکند

حل یک مثال برای start2()

- Present Worth (pw): \$10,000
- Annual Income (ai): \$3,000
- Salvage Value (sv): \$1,000
- Future Worth (fw): \$2,000
- Interest Rate (ir): 5% (0.05)
- Tax Rate (tr): 20% (0.20)
- Project Lifetime (n): 10 years



NPV: \$11289.77
IRR: 23.25%
Payback Period: 4 years
Is the project economical? Yes

مراجع

• ارزیابی پژوههای اقتصادی، محمد مهدی اسکویی نژاد

