**Assignment for CIE-1**

Problem Statement: Develop a Python program to compare two renewable energy projects—solar and wind—using Return on Investment (ROI), Net Present Value (NPV), and Internal Rate of Return (IRR). Based on the financial analysis, the program should recommend the project with the higher NPV.

Values Provided:

Solar Project:

Initial Investment: $200,000

Annual Cash Flow: $50,000

Project Duration: 6 years

Discount Rate: 10% and 15 %

Wind Project:

Initial Investment: $180,000

Annual Cash Flow: $45,000

Project Duration: 6 years

Discount Rate: 10% and 15%

Output:

o Calculate and display ROI, NPV, and IRR for both projects.

o Print a recommendation based on which project has the higher NPV.

Expected Output:

o ROI, NPV, and IRR for Solar and Wind Projects

o Final Recommendation: Select the project with the higher NPV

import numpy as np

import numpy\_financial as npf

def calculate\_roi(initial\_investment, annual\_cash\_flow, duration):

net\_profit = (annual\_cash\_flow \* duration) - initial\_investment

return (net\_profit / initial\_investment) \* 100

def calculate\_npv(initial\_investment, annual\_cash\_flow, duration, discount\_rate):

npv = sum(annual\_cash\_flow / ((1 + discount\_rate) \*\* t) for t in range(1, duration + 1)) - initial\_investment

return npv

def calculate\_irr(initial\_investment, annual\_cash\_flow, duration, discount\_rates):

npv\_r1 = calculate\_npv(initial\_investment, annual\_cash\_flow, duration, discount\_rates[0])

npv\_r2 = calculate\_npv(initial\_investment, annual\_cash\_flow, duration, discount\_rates[1])

r1, r2 = discount\_rates

irr = r1 + (npv\_r1 / (npv\_r1 - npv\_r2)) \* (r2 - r1)

return irr \* 100

def analyze\_project(name, initial\_investment, annual\_cash\_flow, duration, discount\_rates):

print(f"\n-:{name} Project Analysis :-")

roi = calculate\_roi(initial\_investment, annual\_cash\_flow, duration)

irr\_value = calculate\_irr(initial\_investment, annual\_cash\_flow, duration, discount\_rates)

npvs = {rate: calculate\_npv(initial\_investment, annual\_cash\_flow, duration, rate) for rate in discount\_rates}

print(f"--> ROI: {roi:.2f}%")

print(f"--> IRR: {irr\_value:.2f}%")

for rate, npv in npvs.items():

print(f"--> NPV at {rate \* 100:.0f}% Discount Rate: ${npv:.2f}")

return npvs

def main():

*# Given values*

solar = {"name": "Solar", "initial\_investment": 200000, "annual\_cash\_flow": 50000, "duration": 6}

wind = {"name": "Wind", "initial\_investment": 180000, "annual\_cash\_flow": 45000, "duration": 6}

discount\_rates = [0.10, 0.15]

*# Perform analysis*

solar\_npv = analyze\_project(\*\*solar, discount\_rates=discount\_rates)

wind\_npv = analyze\_project(\*\*wind, discount\_rates=discount\_rates)

*# Select best project based on higher NPV at 10% discount rate*

best\_project = "Solar" if solar\_npv[0.10] > wind\_npv[0.10] else "Wind"

print(f"\n--> Final Recommendation : Select the '{best\_project}' project.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

Code with the user input

import numpy as np

import numpy\_financial as npf

def calculate\_roi(initial\_investment, annual\_cash\_flow, duration):

net\_profit = (annual\_cash\_flow \* duration) - initial\_investment

return (net\_profit / initial\_investment) \* 100

def calculate\_npv(initial\_investment, annual\_cash\_flow, duration, discount\_rate):

npv = sum(annual\_cash\_flow / ((1 + discount\_rate) \*\* t) for t in range(1, duration + 1)) - initial\_investment

return npv

def calculate\_irr(initial\_investment, annual\_cash\_flow, duration, discount\_rates):

npv\_r1 = calculate\_npv(initial\_investment, annual\_cash\_flow, duration, discount\_rates[0])

npv\_r2 = calculate\_npv(initial\_investment, annual\_cash\_flow, duration, discount\_rates[1])

r1, r2 = discount\_rates

irr = r1 + (npv\_r1 / (npv\_r1 - npv\_r2)) \* (r2 - r1)

return irr \* 100

def analyze\_project(name, initial\_investment, annual\_cash\_flow, duration, discount\_rates):

print(f"\n-:{name} Project Analysis :-")

roi = calculate\_roi(initial\_investment, annual\_cash\_flow, duration)

irr\_value = calculate\_irr(initial\_investment, annual\_cash\_flow, duration, discount\_rates)

npvs = {rate: calculate\_npv(initial\_investment, annual\_cash\_flow, duration, rate) for rate in discount\_rates}

print(f"--> ROI: {roi:.2f}%")

print(f"--> IRR: {irr\_value:.2f}%")

for rate, npv in npvs.items():

print(f"--> NPV at {rate \* 100:.0f}% Discount Rate: ${npv:.2f}")

return npvs

def get\_project\_input(project\_number):

name = input(f"Enter name for Project {project\_number}: ")

initial\_investment = float(input(f"Enter initial investment for {name}: "))

annual\_cash\_flow = float(input(f"Enter annual cash flow for {name}: "))

duration = int(input(f"Enter duration (years) for {name}: "))

return {"name": name, "initial\_investment": initial\_investment, "annual\_cash\_flow": annual\_cash\_flow, "duration": duration}

def main():

*# Taking user input for both projects*

project1 = get\_project\_input(1)

project2 = get\_project\_input(2)

discount\_rates = [

float(input("Enter first discount rate (as decimal, e.g., 0.10 for 10%): ")),

float(input("Enter second discount rate (as decimal, e.g., 0.15 for 15%): "))

]

*# Perform analysis*

npv1 = analyze\_project(\*\*project1, discount\_rates=discount\_rates)

npv2 = analyze\_project(\*\*project2, discount\_rates=discount\_rates)

*# Select best project based on higher NPV at first discount rate*

best\_project = project1["name"] if npv1[discount\_rates[0]] > npv2[discount\_rates[0]] else project2["name"]

print(f"\n--> Final Recommendation : Select the '{best\_project}' project.")

if \_\_name\_\_ == "\_\_main\_\_":

main()