**🎯 Estrutura do Projeto**

O programa irá apresentar um **menu interativo em Python** (usando input).

**Menu de Opções**

1. **Exercícios resolvidos em PDF**
   * Usuário acessa um arquivo .pdf com exercícios de estatística/probabilidade resolvidos.
   * Você pode gerar esse PDF com reportlab ou simplesmente preparar antes e só abrir pelo programa.
2. **Quiz sobre Estatística em IA**
   * Perguntas de múltipla escolha, com feedback ao usuário.
   * Exemplo: "Qual é a distribuição mais usada para modelar erro em regressão linear?"
3. **Quiz sobre Machine Learning**
   * Perguntas mais voltadas para algoritmos e conceitos (ex.: "O que é overfitting?", "Qual é a função da regularização?").
4. **Pesquisa sobre uso de IA em uma área escolhida**
   * PDF gerado pelo aluno (exemplo: IA na saúde, educação, finanças).
   * Pode ser gerado com reportlab ou entregue pronto e aberto pelo programa.

**🛠 Tecnologias / Bibliotecas em Python**

* **Menus**: print + input para interface gráfica simples.
* **Abrir PDFs**: os.startfile (Windows).
* **Quiz**: usar dicionários/listas para perguntas e respostas, com contagem de acertos.

### ✅ Como usar

1. Salve esse código em um arquivo Python, por exemplo: projeto\_final.py.
2. Coloque os PDFs na mesma pasta do programa:
   * exercicios\_resolvidos.pdf
   * pesquisa\_ia.pdf

### 1. Retorno ao programa principal após abrir PDF

No **Windows**, quando você usa os.startfile("arquivo.pdf"), o Python apenas **abre o PDF com o aplicativo padrão** (geralmente o Adobe Reader ou Edge) e **continua rodando normalmente**.  
Ou seja, o programa **não fica travado** esperando o usuário fechar o PDF — o menu segue rodando, e quando o usuário fechar o arquivo no leitor de PDF, nada é afetado.

Então, o programa funciona assim:

* O usuário abre o PDF.
* Fecha o PDF quando quiser.
* O programa principal continua aberto no terminal.

Sobre os PDF:

 Criar um **PDF de exercícios resolvidos** em Python.

 Esse PDF deve conter **tópicos variados**, como:

1. **Probabilidade clássica** (urna, dados, moeda etc.).
2. **Regressão linear** (relação entre variáveis).
3. **Gráficos** usando numpy e pandas.
4. **Resolução de funções e equações** em Python (ex.: com sympy).
5. Outros exemplos a definir.

Estrutura do PDF sobre exercícios resolvidos:

### Estrutura sugerida para o PDF de exercícios resolvidos

1. **Capa**
   * Título do trabalho (Exercícios Resolvidos de Estatística e Python)
   * Autor (seu nome ou grupo)
   * Curso e instituição (Técnico em IA – Senac)
   * Data
2. **Sumário** (opcional, mas fica bonito).
3. **Capítulo 1 – Probabilidade**
   * Exercício 1: Probabilidade em uma urna (com código Python).
   * Exercício 2: Lançamento de um dado.
   * Exercício 3: Lançamento de moedas.
4. **Capítulo 2 – Regressão Linear**
   * Exercício 4: Regressão linear simples entre duas variáveis (exemplo: horas de estudo vs. nota). Gráfico com matplotlib.
5. **Capítulo 3 – Estatística Descritiva com Pandas**
   * Exercício 5: Média, mediana, moda, variância e desvio padrão de um conjunto de dados.
6. **Capítulo 4 – Resolução de Funções e Equações**
   * Exercício 6: Criar e resolver uma equação do primeiro grau (equação da reta).
   * Exercício 7: Criar e resolver uma equação quadrática.
7. **Capítulo 5 – Visualizações**
   * Exercício 8: Histograma de dados simulados com numpy + matplotlib + seaborn.

✅ Assim, os alunos terão:

* Enunciado
* Código Python
* Saída esperada (quando aplicável)
* Gráfico (quando aplicável)

E cada um pode **formatar isso no PDF manualmente**.