



Começaremos
EM BREVE

Checkpoint 2 – Modelagem Linear para Aprendizado de Máquina

Sistema otimizado na área da saúde

Com Python 

Integrantes:

- Anny Elly Panjota Martin - RM 565055
- Arthur Reis Batista da Silva - RM 562181
- Leonardo de Magalhães Piassa - RM 563663
- Manoela Oliveira Bello - RM 563952

Introdução e Justificativa

Por conta do **avanço tecnológico**, se tornou **essencial otimizar processos** na **área da saúde** para **agilidade** e **eficiência** na hora do atendimento. Esse projeto irá abordar o **desenvolvimento** de um sistema de autoatendimento em um hospital fictício chamado **Sr. Consultas**, usando **Python**, com foco na **organização** e **filtragem inteligente** de pacientes. O intuito é tornar o atendimento mais **humanizado, rápido** e **eficaz**, beneficiando tanto os **pacientes** quanto o **profissional da saúde**.



Introdução ao código

- O sistema foi desenvolvido com **Python** utilizando a biblioteca **pandas**, mais especificamente o **DataFrame**.
- Permite que os pacientes agendem consultas com base em **nome, idade, gênero e data**.
- **Filtra automaticamente** opções inadequadas com base em **gênero e idade**.



A importância da automação em agendamentos médicos

- **Exemplo no código:**
 - **Filtros inteligentes** (ex: bloqueio de Ginecologista para homens e Pediatra para adultos).
 - **Validações automáticas** (idade e gênero).
- **Impacto:**
 - **Redução de erros** com regras pré-definidas (evita agendamentos incorretos).
 - **Autoatendimento** via terminal, liberando equipe administrativa.



Uso de dicionários e pandas

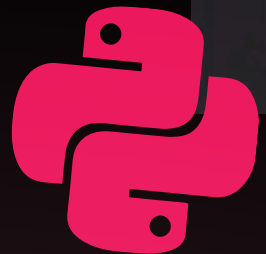
- **Dicionários:**

```
def filtro(nome, idade, genero, data_consulta):  
    # dicionarios  
    precos = {  
        "Clínico Geral": 120.00,  
        "Ginecologista": 350.00,  
        "Odontologista": 200.00,  
        "Pediatra": 180.00,  
        "Dermatologista": 250.00  
    }  
    consultas = {  
        1: "Clínico Geral",  
        2: "Ginecologista",  
        3: "Odontologista",  
        4: "Pediatra",  
        5: "Dermatologista"  
    }
```

- **Pandas:**

```
# criando um dataframe para organizar o resumo da consulta  
df = pd.DataFrame(valores, columns=['Paciente', 'Consulta', 'Data', 'Valor', 'Pagamento'])
```

Transforma agendamentos em
tabela estruturada para
exportação (CSV/Excel).



Boas práticas na interação com o usuário

- **Validações no código:**

- Checagem de gênero/idade para especialidades:

```
if (consulta == 'Ginecologista' and genero == 'M'):
    print('A opção de "Ginecologista" não está disponível para o gênero masculino.')
    continue
```

- Tratamento de erros (try-except para inputs numéricos).

```
except ValueError:
    print('Por favor, digite um número válido.')
```



- Feedback claro:
- Mensagens como "A opção não está disponível para o gênero masculino." (Como mostrado na primeira imagem)
- Confirmação de pagamento (opções S/N).

```
# area de confirmação de pagamenmtto
confirmacao = input('\nDeseja prosseguir?(S/N): ').upper()
```

Análise de dados dos agendamentos

- **Dados coletados:**
- **Nome, tipo de consulta, data da consulta, valor, forma de pagamento** (armazenados em valores[]).

```
# adicionando os valores na lista  
valores.append([nome, consulta, data_consulta, valor_consulta, formas[opcao_pagamento]])
```

- Exemplos de aplicações:
- Financeiro: Calcular receita por especialidade
(df.groupby('Consulta')['Valor'].sum()).
- Logística: Identificar dias com mais agendamentos
(df['Data'].value_counts()).
- Saúde: Mapear demanda por faixa etária (ex: pediatria × adultos).



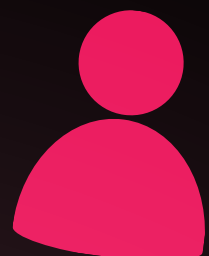
Modificações Futuras

- Trocar as perguntas no terminal por uma tela, ex: site ou aplicativo.
- Guardar os dados em um DataBase seguro, em vez de manter tudo na memória. Protegendo as informações dos pacientes.
- Permitir que o pagamento fosse feito direto pelo sistema, de forma automática e segura.
- Para um sistema mais completo, criar um login para médicos, pacientes e funcionários. Com permissões diferentes para cada um.
- Lembretes por e-mail e SMS sobre a data das consultas.



Conclusão

- **Solução funcional:** O sistema permite que o paciente agende sua consulta com base em filtros personalizados de forma intuitiva.
- **Redução de erros:** Ao restringir especialidades com base na idade e gênero, evitamos escolhas inadequadas e otimizamos o atendimento.
- **Melhoria na experiência do usuário:** Interface simples e direta, com validação de entrada e personalização.
- **Potencial de expansão:** A lógica pode ser adaptada para sistemas reais de clínicas, adicionando banco de dados, autenticação e envio de confirmações automáticas.





OBRIGADO
PELA ATENÇÃO!