🧠 Relatório Pessoal – Trabalho em Grupo de Machine Learning

Nesse trabalho em grupo, a gente colocou em prática várias etapas de um projeto de ciência de dados, com foco em classificação. O objetivo era explorar uma base de dados, entender os padrões e montar um modelo que conseguisse fazer previsões de forma eficiente. Aqui vai um resumo do que foi feito, etapa por etapa, e por que cada parte foi importante no processo.

🔹 1. Começando com as ferramentas certas

Logo de cara, instalamos duas bibliotecas que ajudaram bastante: o **PyCaret** e o **YData Profiling**.

- O PyCaret é tipo um "tudo-em-um" que ajuda a testar vários modelos com poucas linhas de código.
- Já o YData Profiling cria um relatório bem completo da base de dados, com gráficos, estatísticas e tudo que a gente precisa pra entender o que está lidando.

Essas ferramentas economizaram um bom tempo e deixaram tudo mais visual e fácil de interpretar.

🔹 2. Preparando o terreno

Depois disso, importamos as bibliotecas mais clássicas de quem trabalha com dados: pandas, numpy, matplotlib, seaborn e scikit-learn.
Cada uma tem um papel específico: algumas ajudam a manipular os dados, outras a criar gráficos, e outras a construir e avaliar os modelos. É o básico, mas essencial.

🔹 3. Analisando os dados

Com a base carregada, usamos o YData Profiling pra fazer uma **análise exploratória automática**.
Isso mostrou pra gente várias coisas importantes, como:

- Se tinha valor faltando
- Como os dados estavam distribuídos
- Quais variáveis pareciam mais relevantes

Essa parte é tipo conhecer o terreno antes de construir qualquer coisa. Ajuda muito a evitar erros mais pra frente.

🔹 4. Iniciando o ambiente do PyCaret

Aqui usamos o famoso setup() do PyCaret. Com ele, conseguimos preparar os dados pro modelo automaticamente, sem ter que codificar cada passo.
Ele já cuida de coisas como:

- Tratar valores faltantes
- Codificar variáveis categóricas
- Normalizar os dados
- Separar entre treino e teste

Isso foi ótimo pra ganhar tempo e manter o foco no modelo em si.

🔹 5. Comparando os modelos

Com os dados prontos, usamos o compare_models() pra deixar o PyCaret testar vários algoritmos e ver qual funcionava melhor pro nosso caso.
Essa parte foi bem legal porque conseguimos ver quais modelos entregavam os melhores resultados sem ter que montar um por um.

🔹 6. Avaliando a performance

Depois de escolher o melhor modelo, avaliamos como ele estava performando. Usamos:

- **Matriz de confusão**, pra ver os acertos e erros
- **Curva ROC**, pra medir a capacidade de distinguir entre as classes
- **F1-score** e outras métricas pra entender o equilíbrio entre precisão e recall

Essas métricas ajudam a ter certeza de que o modelo tá mandando bem, e não só chutando bonito.

🔹 7. Entendendo o que importa

Por fim, usamos uma técnica chamada **permutação** pra entender quais variáveis mais influenciaram o modelo.

Isso é importante pra deixar o modelo mais explicável e também pra gente saber quais dados realmente fazem diferença na previsão.

✅ Conclusão

Trabalhar em grupo nesse projeto foi uma experiência bem completa. Cada um contribuiu de uma forma, seja pesquisando ferramentas, analisando os dados, ou ajudando a avaliar os modelos.
Acho que o maior aprendizado foi perceber como cada etapa tem seu valor, desde entender os dados até explicar o porquê de o modelo estar funcionando.
No final, conseguimos montar algo funcional, com boas métricas e, principalmente, com um entendimento claro do que estava acontecendo por trás.