Relatório <Número do Card> - <Título do Card>

<Nome do participante>

Descrição da atividade

Este módulo abordou conceitos fundamentais de estatística aplicada ao aprendizado de máquina usando Python. Foi uma experiência ver como essas ferramentas facilitam análises que antes pareciam complexas, mostrando que por trás dos algoritmos de ML existe matemática sólida e bem estabelecida.

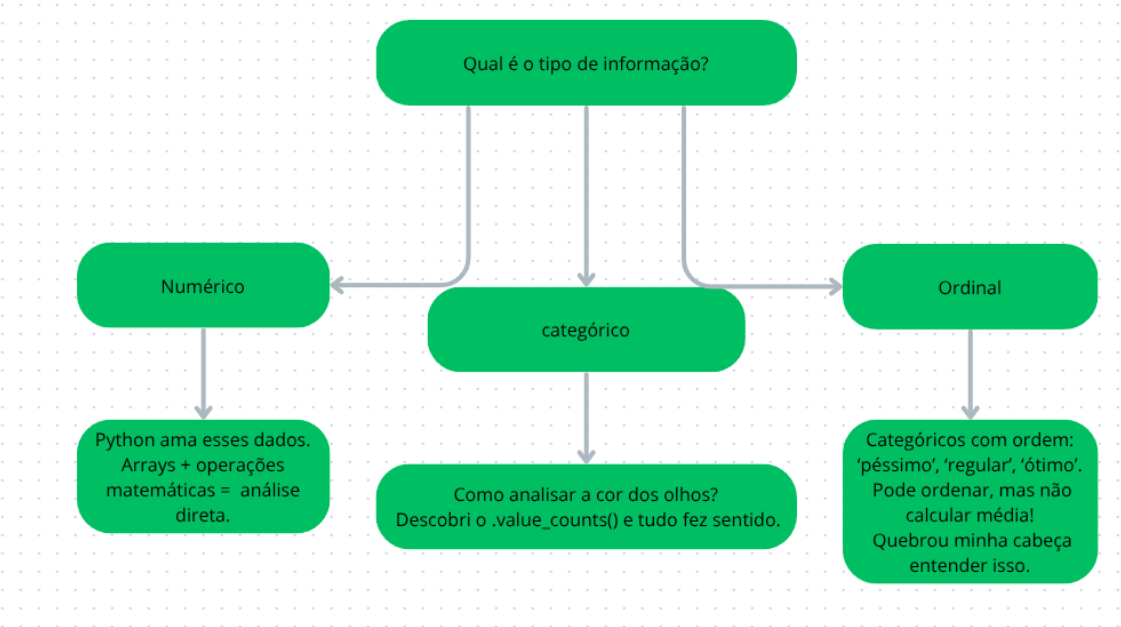
Types of Data (Numerical, Categorical, Ordinal)

Inicialmente subestimei a importância dessa classificação, mas percebi que identificar corretamente o tipo de dado determina completamente quais análises são possíveis.

Dados Numéricos são os mais diretos de trabalhar. Representam quantidades como idade, salário, altura. O Python manipula esses dados facilmente em arrays, permitindo qualquer operação matemática. Dividem-se em discretos (números inteiros como quantidade de filhos) e contínuos (valores infinitos como peso).

Dados Categóricos inicialmente me confundiram. Como analisar "cor dos olhos"? Descobri que o `.value_counts()` resolve essa questão, contando frequências de cada categoria. Mesmo quando codificados numericamente (1 para sim, 2 para não), esses números não têm significado matemático.

Dados Ordinais foram os mais desafiadores. São categóricos mas com ordem hierárquica, como avaliações de "péssimo" a "ótimo". Permitem ordenação mas não cálculos de média, o que quebrou minha cabeça inicialmente.



Mean, Median, Mode

Aqui Python realmente brilhou, automatizando cálculos que eu fazia manualmente no ensino médio.

A **média** é intuitiva: soma tudo e divide pela quantidade. `np.mean()` resolve instantaneamente. Porém descobri que valores extremos distorcem significativamente o resultado.

A **mediana** virou minha medida favorita para dados com outliers. É o valor central dos dados ordenados e não sofre tanto com valores extremos. Em dados salariais onde alguns ganham muito mais, a mediana representa melhor a realidade da maioria.

A **moda** é problemática em Python. Nem sempre existe um valor mais repetido, e `statistics.mode()` às vezes falha em casos de empate. Tive que implementar tratamentos específicos para esses casos.

Variation and Standard Deviation

Esta seção me mostrou que duas distribuições podem ter médias idênticas mas serem completamente diferentes.

A **variância** mede o espalhamento dos dados através da média dos quadrados das diferenças. O problema é que fica em unidade ao quadrado (metros²), dificultando interpretação.

O **desvio padrão**, sendo a raiz da variância, retorna à unidade original e é muito mais interpretável. Desvio baixo indica consistência, alto indica dispersão. Aprendi a usá-lo para identificar outliers: valores que ficam mais de dois desvios padrão da média merecem atenção especial.

Probability Density Function, Probability Mass Function

Os gráficos foram fundamentais para entender essa distinção conceitual.

PMF trabalha com dados discretos como faces de dado ou número de filhos. Visualiza-se com barras, cada uma representando a probabilidade exata de um valor específico.

PDF trata dados contínuos como altura ou peso. Não faz sentido perguntar a probabilidade de alguém ter exatamente 1,753284 metros. Trabalha-se com intervalos e curvas suaves.

Matplotlib esclareceu essa diferença: PMF são barrinhas, PDF são curvas contínuas.

Common Data Distributions (Normal, Binomial, Poisson, etc)

Conectar teoria com mundo real foi revelador. Cada distribuição tem seu lugar específico.

A **distribuição normal** é a famosa curva de sino. Altura, peso, QI seguem esse padrão. O útil é saber que 68% dos dados ficam a um desvio padrão da média. Em Python, `scipy.stats` tem tudo implementado.

Distribuição uniforme tem probabilidades iguais para todos os valores, como um dado honesto.

Distribuição binomial modela situações com dois resultados possíveis em múltiplas tentativas. Quantas caras em 10 lançamentos de moeda? Binomial resolve.

Distribuição de Poisson trata eventos raros: quantos emails por hora, acidentes por semana. Muito específica, mas quando se aplica, é perfeita.

Distribuição exponencial segue "power law", comum em distribuição de renda ou popularidade de sites.

Percentiles and Moments

Percentis são mais úteis que imaginava. Dividem dados ordenados em porcentagens. Percentil 90 significa estar melhor que 90% das pessoas. Vestibulares usam essa lógica.

Momentos quantificam características das distribuições:

- Primeiro momento: média (centralidade)
- Segundo momento: variância (dispersão)
- Terceiro momento: assimetria (se a curva pende para um lado)
- Quarto momento: curtose (se a curva é pontiaguda ou achatada)

Na prática, uso principalmente os dois primeiros. Os outros são mais para pesquisa acadêmica.

A Crash Course in matplotlib e Advanced Visualization with Seaborn

Matplotlib tem sintaxe estranha inicialmente. Oferece `plt.plot()` para simplicidade e versão orientada a objetos com `fig, ax = plt.subplots()`

para controle total. A customização impressiona: posso mudar cores, estilos, títulos, legendas. Os gráficos ficam profissionais.

Seaborn salvou minha vida. Pega matplotlib e deixa tudo mais bonito automaticamente. `sns.histplot()` produz histogramas superiores com uma linha. `sns.pairplot()` é genial: mostra relacionamentos entre todas as variáveis numa matriz, perfeito para exploração inicial de datasets.

Covariance and Correlation

Importante para não cair em pegadinhas estatísticas.

Covariância mede se duas variáveis se movem juntas, mas o valor é difícil de interpretar diretamente.

Correlação é superior: padroniza covariância dividindo pelos desvios padrão, resultando em valores entre -1 e +1. Próximo de +1 significa que quando uma sobe, a outra sobe. Próximo de -1 significa movimento oposto. Próximo de 0 indica ausência de relacionamento linear.

Crucial lembrar: correlação não é causalidade. Vendas de sorvete e afogamentos correlacionam porque ambos aumentam no verão, não porque sorvete causa afogamento.

`sns.heatmap()` com matriz de correlação fica visualmente impactante.

Conditional Probability

Probabilidade condicional P(A|B) difere completamente de P(A). É a chance de A dado que B já aconteceu.

Exemplo prático: se 60% dos estudantes passaram em ambos os testes e 80% passaram no primeiro, qual a chance de passar no segundo dado que passou no primeiro? P(segundo|primeiro) = 0,6/0,8 = 75%.

Em Python, uso filtering de dataframes para implementar essas análises condicionais em dados reais.

Bayes' Theorem

O Teorema de Bayes quase fritou meu cérebro. Inverte probabilidades usando: P(A|B) = P(A) × P(B|A) / P(B).

O exemplo clássico de teste médico é contraintuitivo: mesmo com 99% de precisão testando uma condição rara (0,3% da população), se você testa positivo, a chance real de ter a condição é apenas cerca de 23%, não 99%.

Isso acontece porque a probabilidade base (prevalência) afeta drasticamente a interpretação. Matemática não mente, nossa intuição é que falha.

Conclusões

Este módulo mudou minha perspectiva sobre estatística. Python transformou cálculos tediosos em análises poderosas. Ver distribuições em gráficos matplotlib/seaborn fez conceitos abstratos se tornarem tangíveis.

Entender tipos de dados é fundamental: categóricos servem para segmentação, numéricos permitem cálculos completos. Medidas de tendência central revelam aspectos diferentes: mediana é mais robusta que média quando há outliers.

Percentis orientam decisões práticas. Saber que 75% das vendas ficam abaixo de determinado valor direciona estratégias comerciais. Correlação revela relacionamentos entre variáveis, mas sempre lembrando que correlação não implica causalidade.

Probabilidade condicional e Bayes fundamentam raciocínio estatístico avançado, essencial para interpretar corretamente testes e fazer previsões confiáveis.

Agora vejo aplicações em tudo: segmentação de clientes usa clustering, filtros de spam aplicam Bayes, propagandas direcionadas exploram correlações. É como ganhar óculos especiais para enxergar a matemática por trás do mundo digital.

Referencias

Material do curso: Estatística para Aprendizado de Máquina (I)

Notebooks e exercícios práticos realizados durante o módulo