



**Centro Universitário de Excelência
Sistemas de Informação**

Mineração de Dados: Projeto Tia Lu Delivery

autores: Herick Brito
Juary Júnior
Lucas Lan

Feira de Santana, 2025

Agenda

Objetivo da apresentação

Uso da linguagem python para criação do projeto FoodDelivery mostrando fundamentos usados operações estatísticas, funcionalidades, estruturas, aprendizados e finalidades do projeto.

1. Introdução

Contextualização da importância do conhecimento em estatística

2. Fundamentação Teórica

Observação e explicação das estruturas e códigos usados para as operações estatísticas

3. Metodologia

formas e métodos que usamos a estrutura de dados a formação e a conclusão do nosso obje

4. Resultados e Desafios

Funcionalidade do projeto e dificuldades durante o processo

5. Referências

Conhecimentos obtidos através do projeto

1. Introdução

- Vivemos na era do Big Data, com grandes volumes de informações gerados constantemente.
- A Estatística é essencial para coletar, organizar, analisar e interpretar dados.
- O valor está em extrair informações, padrões e insights.
- Permite transformar números complexos em resumos claros e conclusões confiáveis.
- **Objetivo:** aplicar teoria na prática, implementando algoritmos manuais para compreender seu funcionamento, sem uso de bibliotecas prontas.

2. Fundamentação Teórica



Operações estatísticas:

- **Média:** valor central de um conjunto de dados (ex: tempo médio de entrega de pedidos).
- **Moda:** valor mais recorrente (ex: método de pagamento mais usado pelos clientes).
- **Mediana:** ponto que divide os dados ao meio (ex: metade dos clientes gasta até R\$ 50).
- **Variância e Desvio Padrão:** medem a dispersão dos dados (ex: diferença de notas em avaliações de restaurantes).
- **Covariância:** mostra como duas variáveis se relacionam (ex: aumento do valor do pedido conforme cresce o número de itens).
- **Frequências:**
 - **Absoluta** → contagem direta dos valores (ex: 120 pedidos pagos em dinheiro).
 - **Relativa** → proporção em relação ao total (ex: 40% dos pedidos foram via cartão).
 - **Acumulada** → soma progressiva das ocorrências (ex: até 200 pedidos, 150 foram de pratos principais).
- **Itens Únicos:** lista de valores distintos (ex: número de bairros diferentes atendidos).
- **Probabilidade Condicional:** chance de um evento ocorrer dado que outro já ocorreu (ex: probabilidade de um cliente pedir bebida após escolher uma pizza).

Método Construtor



```
def __init__(self, dataset):  
    """  
    Inicializa o objeto Statistics.  
  
    Parâmetros  
    -----  
    dataset : dict[str, list]  
        O conjunto de dados, onde as chaves representam os nomes das  
        colunas e os valores são as listas de dados correspondentes.  
    """  
  
    if not isinstance(dataset, dict):  
        raise TypeError("O dataset deve ser um dicionário.")  
  
    for valor in dataset.values():  
        if not isinstance(valor, list):  
            raise TypeError("Todos os valores no dicionário do dataset devem ser listas.")  
  
    if dataset:  
        lista_valores = list(dataset.values() )  
        tamanho_lista_valores_referencia = len(lista_valores[0])  
        for lista in lista_valores[1:]:  
            if len(valor) != tamanho_lista_valores_referencia:  
                raise ValueError("Todas as colunas no dataset devem ter o mesmo tamanho.")
```

Mediana



```
if column not in self.dataset:
    raise KeyError(f"A coluna '{column}' não existe no dataset.")

valores_coluna = self.dataset[column]

if not valores_coluna:
    return 0.0

quantidade_valores = len(valores_coluna)

for i in range(quantidade_valores):
    for j in range(0, quantidade_valores - i - 1):
        if valores_coluna[j] > valores_coluna[j + 1]:
            valores_coluna[j], valores_coluna[j + 1] = valores_coluna[j + 1], valores_coluna[j]

if quantidade_valores % 2 == 0:
    indice = int(quantidade_valores / 2)
    mediana = (valores_coluna[indice-1] + valores_coluna[indice]) / 2
else:
    indice_meio = int(quantidade_valores / 2)
    mediana = valores_coluna[indice_meio]

return mediana
```

Moda



```
if column not in self.dataset:
    raise KeyError(f"A coluna '{column}' não existe no dataset.")

valores_coluna = self.dataset[column]

if not valores_coluna:
    return []

contador = {}
for valor in valores_coluna:
    if valor in contador:
        contador[valor] += 1
    else:
        contador[valor] = 1

numero_maximo_repeticoes = max(contador.values())

modas = []
for chave, valor in contador.items():
    if(valor == numero_maximo_repeticoes):
        modas.append(chave)

return modas
```

Covariância



```
if column_a not in self.dataset:
    raise KeyError(f"A coluna a: {column_a} não existe no dataset")

if column_b not in self.dataset:
    raise KeyError(f"A coluna b: {column_b} não existe no dataset")

valores_coluna_a = self.dataset[column_a]
valores_coluna_b = self.dataset[column_b]

if not valores_coluna_a:
    return 0.0

media_coluna_a = self.mean(column_a)
media_coluna_b = self.mean(column_b)

soma = 0
contador = 0
for valor_a in valores_coluna_a:
    produto = (valor_a - media_coluna_a) * (valores_coluna_b[contador] - media_coluna_b)

    contador += 1
    soma += produto

quantidade_pares = len(valores_coluna_a)

covariancia = soma / quantidade_pares

return covariancia
```


Referências

DEVMEDIA. Mineração de Dados Educacionais usando KDD – Parte 1. Disponível em: <https://www.devmedia.com.br/mineracao-de-dados-educacionais-usando-kdd-parte-1/28968> . Acesso em: 23 ago. 2025.

DATA AVENUE – Desenvolvimento de Dados. Principais métricas estatísticas utilizadas em ciência e análise de dados. Disponível em: <https://dataavenue.com.br/blog/principais-metricas-estatisticas-utilizadas-em-ciencia-e-analise-de-dados/> . Acesso em: 23 ago. 2025.

LINKAGES. Dados: quantos geramos e como eles se transformam em insights. Disponível em: <https://linkages.com.br/2023/03/29/dados-quantos-geramos-e-como-isso-impacta-nossa-vida/> Acesso em: 23 ago. 2025.