Documentação do Processo de Análise de Dados

Autor: Andrey de Oliveira Sabino

Orientador: Alysson Filgueira Milanez

Finalidade: Este projeto de pesquisa e analise tem como objetivo contribuir com o ensino de programação, através da análise de dados de frequência e desempenho dos alunos, pretende-se analisar os impactos que as ações de extensão tem no ensino e eficiência dos discentes.

Organização do Diretório

Os dados e scripts estão organizados da seguinte forma:

```
DATAFRAMES/
├── 201801/
│   ├── 1_conversao_inicial.py
│   ├── 2_conversao_secundaria.py
│   ├── 3_type.py
│   ├── 4_PEAR.py
│   ├── 5_DISP.py
│   ├── 2001801-VF.csv
├── 201802/
│   ├── 1_conversao_inicial.py
│   ├── 2_conversao_secundaria.py
│   ├── 3_type.py
│   ├── 4_PEAR.py
│   ├── 5_DISP.py
│   ├── 201802-VF.csv
├── 201901/
│   ├── 1_conversao_inicial.py
│   ├── 2_conversao_secundaria.py
│   ├── 3_type.py
│   ├── 4_PEAR.py
│   ├── 5_DISP.py
│   ├── 201901-VF.csv
├── 201902/
│   ├── 1_conversao_inicial.py
│   ├── 2_conversao_secundaria.py
│   ├── 3_type.py
│   ├── 4_PEAR.py
│   ├── 5_DISP.py
│   ├── 201902-VF.csv
├── 202001/
│   ├── 1_conversao_inicial.py
│   ├── 2_conversao_secundaria.py
│   ├── 3_type.py
│   ├── 4_PEAR.py
│   ├── 5_DISP.py
│   ├── 202001-VF.csv
```

Cada pasta dentro do diretório `DATAFRAMES` corresponde a um semestre e contém os arquivos CSV resultantes de cada etapa do processo (conversão, limpeza e normalização), bem como os scripts utilizados para cada etapa.

- `1_conversao_inicial.py`: Script para a conversão inicial dos dados.
- `2_conversao_secundaria.py`: Script para a segunda fase de conversão dos dados.
- `3_type.py`: Script para verificar os tipos de dados.
- `4_PEAR.py`: Script para calcular o coeficiente de correlação de Pearson.
- `5_DISP.py`: Script para gerar gráficos de dispersão.
- `2001801-VF.csv`: Arquivo CSV resultante da conversão inicial dos dados.

1. Introdução

Este documento detalha o processo de análise de dados realizado no projeto de TCC, desde a conversão dos dados originais em formato `.pdf` para `.xlsx` e `.csv`, até a limpeza, normalização e preparação dos dados para a análise final. O objetivo é explicar o passo a passo seguido e os desafios enfrentados ao longo do processo.

2. Conversão dos Arquivos PDF para XLSX/CSV

Os dados originais estavam em arquivos PDF, o que dificultava a manipulação direta dos mesmos. Para converter esses arquivos em formatos que permitissem uma análise mais flexível, utilizei o script Python abaixo:

```
import tabula
import pandas as pd
from google.colab import files

# Carregar o arquivo PDF
uploaded = files.upload()
pdf_file = list(uploaded.keys())[0]

# Extrair todas as tabelas do PDF
tables = tabula.read_pdf(pdf_file, pages='all', multiple_tables=True, guess=True,
stream=True)

# Salvar cada tabela como um arquivo Excel e CSV
for i, tabela in enumerate(tables):
  xlsx_file = f"tabela_{i+1}.xlsx"
  csv_file = f"tabela_{i+1}.csv"
  tabela.to_excel(xlsx_file, index=False)
  tabela.to_csv(csv_file, index=False)
  print(f"Tabela {i+1} salva como {xlsx_file} e {csv_file}")
  files.download(xlsx_file)
  files.download(csv_file)
```

3. Limpeza e Redução de Colunas

Após a conversão dos arquivos PDF para formatos manipuláveis como `.xlsx` e `.csv`, iniciou-se a fase de limpeza dos dados. Esta fase incluiu a remoção de colunas irrelevantes, tratamento de dados ausentes e padronização de formatos.

Fase Inicial de Limpeza

1. **Remoção de Colunas Irrelevantes**: Muitas colunas presentes nos dados originais não eram necessárias para a análise. Essas colunas foram removidas para simplificar o conjunto de dados.
2. **Tratamento de Dados Ausentes**: Dados ausentes foram identificados e tratados. Dependendo do caso, valores ausentes foram preenchidos com a média da coluna ou removidos.
3. **Padronização de Formatos**: Garantir que todos os dados estavam no formato correto (por exemplo, datas no formato `YYYY-MM-DD`, números como floats ou inteiros conforme necessário).

Formato dos Dados Antes da Limpeza

Antes da limpeza, os dados tinham o seguinte formato:

ID	06/07/202X	07/07/202X	08/07/202X	09/07/202X	10/07/202X	11/07/202X	12/07/202X	...
1	V	F	V	V	V	V	F	...
2	V	V	F	F	V	V	V	...

- **V** indica presença, e **F** indica falta.

Formato dos Dados Após a Limpeza

Após a limpeza, as colunas menos relevantes foram removidas. Como o conjunto de dados se referia à frequência dos alunos, e cada aula gerava uma nova coluna, isso resultava em um excesso de colunas com a presença ou ausência por data.

Para simplificar, optei por representar a frequência dos alunos em forma de porcentagem, indicando a quantidade total de aulas frequentadas. Além disso, incluímos a **média do aluno na disciplina de Algoritmos** e o **status de aprovação**. Com essas alterações, o conjunto de dados ficou assim:

ID	Nome do Aluno	Frequência (%)	Média em Algoritmos	Status
1	Aluno A	85%	7.5	Aprovado
2	Aluno B	75%	6.0	Aprovado
3	Aluno C	100%	5.0	Reprovado

Essa abordagem oferece uma visão mais resumida e fácil de entender, além de permitir que a análise seja focada não só na frequência, mas também no desempenho dos alunos na disciplina. Dessa forma, ficou mais simples comparar o impacto da frequência no rendimento acadêmico e no status final de aprovação.

Script de Conversão e Limpeza Inicial

```
import pandas as pd

# Carregar os dados convertidos
df = pd.read_csv('dados_convertidos.csv')

# Remover colunas irrelevantes
df = df.drop(columns=['ColunaIrrelevante1', 'ColunaIrrelevante2'])

# Tratar dados ausentes
df = df.fillna(df.mean())

# Padronizar formatos
df['Data'] = pd.to_datetime(df['Data'], format='%Y-%m-%d')
df['Valor'] = df['Valor'].astype(float)

# Salvar os dados limpos
df.to_csv('dados_limpos.csv', index=False)
```

4. Normalização dos Dados

Após a limpeza inicial, foi realizado um processo de normalização para garantir a consistência e facilitar a análise. O processo final resultou em uma tabela com três colunas: `ID`, `MEDIA`, e `PE`.

Representação da Tabela Final

ID	MEDIA	PE
1	7.5	85.0
2	6.0	75.0
3	5.0	100.0

- **ID**: Número inteiro identificador do aluno.
- **MEDIA**: Média dos alunos na disciplina de Algoritmos (float).
- **PE**: Percentual na extensão, que é a frequência na extensão (float).

Scripts Utilizados

Script de Conversão Inicial

```
import pandas as pd

# Carregar os dados convertidos
df = pd.read_csv('dados_convertidos.csv')

# Normalizar os dados
df['MEDIA'] = df['MEDIA'].astype(float)
df['PE'] = df['PE'].astype(float)
df['ID'] = df['ID'].astype(int)

# Salvar os dados normalizados
df.to_csv('dados_normalizados.csv', index=False)
```

Script de Conversão Secundária

```
import pandas as pd

# Carregar os dados convertidos
df = pd.read_csv('dados_convertidos.csv')

# Realizar a segunda fase de conversão
# (Adicionar aqui a lógica específica da segunda fase de conversão)

# Salvar os dados convertidos
df.to_csv('dados_convertidos_secundaria.csv', index=False)
```

Script de Verificação de Tipos

```
import pandas as pd
```

```
# Carregar os dados normalizados
df = pd.read_csv('dados_normalizados.csv')

# Verificar os tipos de dados
print(df.dtypes)
```

Script para Calcular o Coeficiente de Pearson

```
import pandas as pd

# Carregar os dados normalizados
df = pd.read_csv('dados_normalizados.csv')

# Calcular o coeficiente de correlação de Pearson
correlation = df['MEDIA'].corr(df['PE'])
print(f'Coeficiente de Correlação de Pearson: {correlation}')
```

Script para Geração de Gráficos de Dispersão

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Carregar os dados normalizados
df = pd.read_csv('dados_normalizados.csv')

# Gerar gráfico de dispersão
sns.scatterplot(x='MEDIA', y='PE', data=df)
plt.title('Gráfico de Dispersão: MEDIA vs PE')
plt.xlabel('MEDIA')
plt.ylabel('PE')
plt.show()
```

5. Considerações Sobre a Fase de Limpeza e Normalização

O processo de conversão, limpeza e normalização de dados foi uma etapa crucial para garantir a integridade dos dados utilizados no estudo. Embora tenha havido problemas durante a conversão dos arquivos PDF, todos os erros foram identificados e corrigidos a tempo, garantindo a consistência dos dados.

A partir dos arquivos normalizados, foi possível proceder com a análise de dados, focando nos elementos mais relevantes para os objetivos do TCC.

Diagrama do Fluxo de Execução

```
graph LR
A[Conversão dos Arquivos PDF para XLSX/CSV] -->
B[Tratamento de Dados Ausentes] -->
C[Padronização de Formatos] -->
```

```
  D[Redução de Colunas] -->
  E[Normalização dos Dados]
```

Essa abordagem oferece uma visão mais resumida e fácil de entender, além de permitir que a análise seja focada não só na frequência, mas também no desempenho dos alunos na disciplina. Dessa forma, ficou mais simples comparar o impacto da frequência no rendimento acadêmico e no status final de aprovação.