INSTITUTO FEDERAL
Brasília
Campus Brasília

TECNOLOGIA EM SISTEMAS PARA INTERNET

Vitor Rodrigues Ferreira
Nínive Helen Horácio da Silva

RELATÓRIO DE PRÁTICA INTEGRADA DE CIÊNCIA DE DADOS E APRENDIZADO DE MÁQUINA

Brasília - DF

28/01/2022

Sumário

1. Objetivos

O desenvolvimento da sprint 1 tem como foco principal a coleta dos dados sobre movimentos que são transmitidos através de coordenadas. Além da coleta foi aplicado etapas de exploração para uma interpretação melhor dos dados coletados e a parte da preparação, criando variáveis para facilitar uma análise dos dados.

2. Descrição do problema

Conforme descrito na seção anterior, tivemos respectivamente as etapas de coleta, exploração e preparação dos dados de interesse.

Os dados para os diversos movimentos estudados estão registrados em arquivos de texto simples, separados por gênero, numeração do voluntário e instante de tempo, tendo como conteúdo os valores para as coordenadas X, Y e Z, gerados pelo acelerômetro acoplado durante a realização do então movimento. Assim, parte dos dados estava contida no próprio nome do arquivo e outras no corpo dele.

O desafio da parte da coleta foi iterar sobre todas as pastas que continham os registros desses movimentos, e extrair de cada arquivo os dados de ambas as "fontes", separando em estruturas que permitiriam criar posteriormente, um único DataFrame com todos eles.

A etapa de exploração consistiu em obter diversas métricas sobre os dados coletados, como distribuições das medidas, visualizações de registros por gênero e de correlação entre variáveis. Tendo os dados em um contexto de mais fácil manipulação, foram utilizadas as libs do matplotlib e seans para auxiliar na extração de informações de forma visual

Finalmente, compondo a etapa de preparação, teríamos a remoção de dados julgados irrelevantes e também a criação de uma nova variável, a média das coordenadas x, y e z, visando uma maior facilidade numa futura análise. O que foi alcançado com operações simples sobre o DataFrame construído.

3. Desenvolvimento

Em cada etapa do projeto foi utilizado o ambiente de desenvolvimento online do Google o Google Colab, uma ferramenta online e gratuita bastante utilizada para análise de dados e visualização de dados. Na etapa de coleta foi utilizado um link do google drive para fazer a importação dos dados, após esta etapa foi utilizado a biblioteca pandas, para selecionar as tabelas que seriam coletadas. Na etapa de exploração dos dados foi utilizado bibliotecas de visualização de dados, o matplotlib e o seaborn, para fazer gráficos que possibilitem entender melhor a distribuição dos valores. Na etapa de preparação foi utilizado o pandas para criar novas colunas de dados que seriam necessários para uma análise dos dados.

3.1 Código implementado

Coleta de Dados

```
#Importando bibliotecas
 from pandas import DataFrame, Series, concat
 import os, time
 import numpy

 #Função para importa os dados
 def fetch_raw_data():
                                                    os.system("wget-Odata.zip
https://drive.google.com/u/0/uc?id=1xdHaCrNs9sLbO5otCcKoR2TsO2-Uhhnv&export=download"
)
   time.sleep(3)
  os.system("unzip data.zip")
#Chamada da função
fetch_raw_data()

#Função para coletar os dados da pasta
def extract_data_from(filename):
 entry = {}
```

```
  meta = filename[14: -4]
  entry["date"] = meta[0: 19]

  meta = meta[20:].split("-")
  entry["activity"] = meta[0]
  entry["gender"] = meta[1][0]
  entry["number"] = meta[1][1]
  return entry

#Função para coleta dos dados
def create_dataframe():
  base_dir = "HMP_Dataset"
  dfs = []

#Selecionando as pastas
  folders = ["Brush_teeth", "Climb_stairs", "Comb_hair", "Descend_stairs",
             "Drink_glass", "Eat_meat", "Eat_soup", "Getup_bed",
             "Liedown_bed", "Pour_water", "Sitdown_chair",
             "Standup_chair", "Use_telephone", "Walk"] # já sem os do tipo MODEL

#Guardando os valores selecionados em lista
  for folder in folders:
    for filename in os.listdir(f'{base_dir}/{folder}'):
      x_axis, y_axis, z_axis = [], [], []
      entry_data = extract_data_from(filename)

      file1 = open(f'{base_dir}/{folder}/{filename}', 'r')

      for line in file1.readlines():
        values = line.strip().split(" ")
        x_axis.append(values[0])
        y_axis.append(values[1])
        z_axis.append(values[2])

      entry_data["x_axis"] = Series(x_axis).astype(float)
      entry_data["y_axis"] = Series(y_axis).astype(float)
      entry_data["z_axis"] = Series(z_axis).astype(float)

      dfs.append(DataFrame(entry_data))
      break
```

```
 #Função para criar o data frame dos dados selecionados
 df_merged = concat(dfs)
  return df_merged

create_dataframe()
```

Exploração

```
#importando biblioteca matplotlib
import matplotlib.pyplot as plt
import seaborn as sns


#Entendo melhor os valores das variáveis numéricas. Suas métricas
df.describe()

#Gráfico quantidade género
fig, ax = plt.subplots()
#criando o gráfico de barras
sns.barplot(x=df.index, y=df['gender'], ax=ax, palette ="husl",data=df)
#otimizar espaço da figure
fig.tight_layout();

#Correlaçao Entre as variaveis x_axis,  y_axis,z_axis,axis_avg. Tabela
df[['x_axis',       'y_axis','z_axis','axis_avg']].corr()

#Correlação das variaveis
#valores negativos indicam correlação inversa, isto é, quando um cresce o outro diminui;
#positivos indicam crescimento uniforme, ou seja, ambas crescem ou ambas diminuem; os valores vão
de -1 a +1 e
#quanto mais próximo do valor absoluto 1, mais forte a ligação entre as duas variáveis).
#Perceba que a correlação de uma variável com ela mesma sempre será 1.
correlacao =df[['x_axis','y_axis','z_axis','axis_avg']].corr()
correlacao
#Matriz de correlação
plot = sns.heatmap(correlacao, annot = True, fmt=".1f", linewidths=.6)
plot

#Gráfico de histograma das variações da variavel X
x = df['x_axis']
plt.figure(figsize=(8, 6))
```

```
plt.hist(x, bins=range(40, 110,10))
plt.title('Distribuição das medidas  X')
plt.xlabel('X')
plt.ylabel('distribuição')

#Gráfico Tipo de movimento
fig, ax = plt.subplots()
#criando o gráfico de barras
sns.barplot(x=df.index, y=df['activity'], ax=ax, palette ="husl",data=df)
#otimizar espaço da figure
fig.tight_layout();
```

Preparação

```
# Nova coluna de Média entre eixos => axis_avg
df = create_dataframe()
df['axis_avg'] = df.apply(lambda df: numpy.average([df.x_axis, df.y_axis, df.z_axis]), axis=1)
df

# Correlação entre eixos e axis_avg
for axis in ['x', 'y', 'z']:
  corr = df[f'{axis}_axis'].corr(df['axis_avg'])
  print(f'A correlação entre o eixo {axis} e a média entre eixos é de: {corr}')
```

4. Considerações finais

Trabalhando em equipe foi possível fazer as etapas sem maiores dificuldades, e chegamos a resultados satisfatórios, em questão de performance (na coleta de dados), na forma como os dados estão estruturados e nas visualizações geradas, que mostram várias informações relevantes sobre os dados de interesse.

As ferramentas oferecem recursos bastante convenientes e possibilitaram uma maior agilidade ao lidar com várias das exigências para esta sprint. Enxergamos positivamente a exposição à elas e às suas documentações.

Referências

PANDAS, documentação, Disponivel em; <https://pandas.pydata.org/>
MATPLOTLIB, documentação Disponivel em; <https://matplotlib.org/>
SEABORN, documentação, Disponivel em; <https://seaborn.pydata.org/>