Alimentando Informações: Um Estudo De Dados Nutricionais

Alunos :
Silas de Souza Ferreira
RA: 10414793
Israel Soares do N. Viana
RA: 10414894
Gustavo Silva Rios
RA : 10415824

STORYTELLING

Nosso projeto surge como uma resposta prática e estratégica aos desafios enfrentados no pós pandemia, quando os dados nutricionais passaram a evidenciar tanto a persistência da desnutrição em regiões vulneráveis quanto o avanço silencioso do sobrepeso e da obesidade inclusive entre crianças.

A ausência de acompanhamento regular e a fragmentação das informações em sistemas como o Sisvan Web, e-SUS APS e Auxílio Brasil dificultam a identificação de padrões e a execução de ações coordenadas, especialmente onde a insegurança alimentar é mais severa.

A solução que propomos integra diferentes fontes de dados em uma plataforma interativa, com painéis visuais e análises preditivas que facilitam a tomada de decisão por gestores públicos e profissionais da saúde. Ao transformar dados dispersos em inteligência acionável, nosso projeto fortalece a vigilância nutricional e permite intervenções mais ágeis e direcionadas, promovendo uma articulação eficiente entre saúde, assistência social e educação.

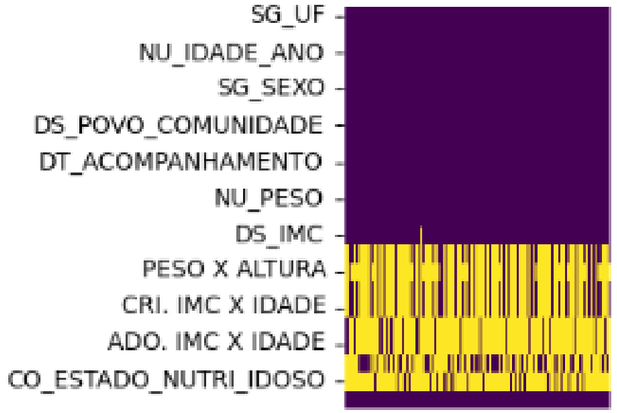
Leitura e Pré-Processamento dos Dados

Merge - Consolidação das Bases

```
lista_dfs = [df_2023, df_2022, df_2021, df_2020, df_2019, df_2018]
df_consolidado = pd.concat(lista_dfs, ignore_index=True)
```

```
df_consolidado.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 212343 entries, 0 to 212342
Data columns (total 22 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   SG_UF                   212343 non-null  object
 1   NO_MUNICIPIO            212343 non-null  object
 2   NU_IDADE_ANO            212343 non-null  int64
 3   DS_FASE_VIDA            212343 non-null  object
 4   SG_SEXO                 212343 non-null  object
 5   DS_RACA_COR             212343 non-null  object
 6   DS_POVO_COMUNIDADE      212343 non-null  object
 7   DS_ESCOLARIDADE         212343 non-null  object
 8   DT_ACOMPANHAMENTO       212343 non-null  object
 9   NU_COMPETENCIA          212343 non-null  int64
 10  NU_PESO                 212343 non-null  object
 11  NU_ALTURA               212343 non-null  object
 12  DS_IMC                  211801 non-null  object
 13  PESO X IDADE            56828 non-null   object
 14  PESO X ALTURA           32085 non-null   object
 15  CRI. ALTURA X IDADE     56819 non-null   object
 16  CRI. IMC X IDADE        56779 non-null   object
 17  ADO. ALTURA X IDADE     31039 non-null   object
 18  ADO. IMC X IDADE        31018 non-null   object
 19  CO_ESTADO_NUTRI_ADULTO  97413 non-null   object
 20  CO_ESTADO_NUTRI_IDOSO   25678 non-null   object
 21  SISTEMA_ORIGEM_ACOMP    212343 non-null  object
dtypes: int64(2), object(20)
memory usage: 35.6+ MB
```



- As **bases de dados** foram adquiridas do site do Ministério da Saúde (SISVAN) referentes aos anos de **2018 a 2023**.
- Os arquivos .csv foram lidos individualmente com a biblioteca *pandas* e **consolidados em uma só base utilizando sua função *.concat()***
- A **base consolidada possui 212.343 registros válidos** e abrange um conjunto robusto de **variáveis demográficas, antropométricas e nutricionais**, coletadas de 2018 a 2023.
- Apesar do volume expressivo, **há ausência de dados em variáveis específicas**, como classificações nutricionais por faixa etária e indicadores clínicos, **exigindo tratamento prévio para análises preditivas**. As variáveis apresentam **tipagens heterogêneas**, exigindo transformação de tipos para modelagem.

Leitura e Pré-Processamento dos Dados

Convertendo variável temporal para formato Datetime

```
df_consolidado['DT_ACOMPANHAMENTO'] = pd.to_datetime(df_consolidado['DT_ACOMPANHAMENTO'], dayfirst=True, errors='coerce')
```

```
df_consolidado['NU_PESO'] = df_consolidado['NU_PESO'].str.replace(',', '.').astype(float)
df_consolidado['NU_ALTURA'] = df_consolidado['NU_ALTURA'].str.replace(',', '.').astype(float)
df_consolidado['DS_IMC'] = df_consolidado['DS_IMC'].str.replace(',', '.').astype(float)
```

- Valores de Idade até 168 anos foram encontrados o que é estatisticamente improvável.

```
# Quantidade de registros com idade maior que 120
outliers_idade = df_consolidado[df_consolidado['NU_IDADE_ANO'] > 120]
print(f"Quantidade de registros com idade > 120: {len(outliers_idade)}")

# Visualizar exemplos (opcional)
outliers_idade[['NO_MUNICIPIO', 'NU_IDADE_ANO', 'DS_FASE_VIDA', 'DT_ACOMPANHAMENTO']].head()
```

Verificação de inconsistências

```
# Verificações baseadas em impossibilidades absolutas (independente da idade)
peso_invalidos = df_consolidado[df_consolidado['NU_PESO'] <= 0]
altura_invalidos = df_consolidado[df_consolidado['NU_ALTURA'] <= 0]
peso_max_invalidos = df_consolidado[df_consolidado['NU_PESO'] > 200]  # excessivo mesmo para adultos
altura_max_invalidos = df_consolidado[df_consolidado['NU_ALTURA'] > 220]
imc_invalidos = df_consolidado[(df_consolidado['DS_IMC'] < 10) | (df_consolidado['DS_IMC'] > 60)]

print(f"Peso zero ou negativo: {len(peso_invalidos)}")
print(f"Altura zero ou negativa: {len(altura_invalidos)}")
print(f"Peso > 200kg: {len(peso_max_invalidos)}")
print(f"Altura > 220cm: {len(altura_max_invalidos)}")
print(f"IMC fora de faixa (10 a 60): {len(imc_invalidos)}")
```

```
Peso zero ou negativo: 411
Altura zero ou negativa: 419
Peso > 200kg: 43
Altura > 220cm: 21
IMC fora de faixa (10 a 60): 658
```

```
df_consolidado = df_consolidado[
    (df_consolidado['NU_PESO'] >= 1.5) & (df_consolidado['NU_PESO'] <= 200) &
    (df_consolidado['NU_ALTURA'] >= 40) & (df_consolidado['NU_ALTURA'] <= 220) &
    (df_consolidado['DS_IMC'] >= 10) & (df_consolidado['DS_IMC'] <= 60)
]
```

- Inicialmente, os campos numéricos NU_PESO, NU_ALTURA e DS_IMC **foram padronizados**, **convertendo valores decimais** com vírgulas para o formato de ponto (.) e **transformando-os em tipo numérico** (float), garantindo a integridade para análises estatísticas e modelagens.

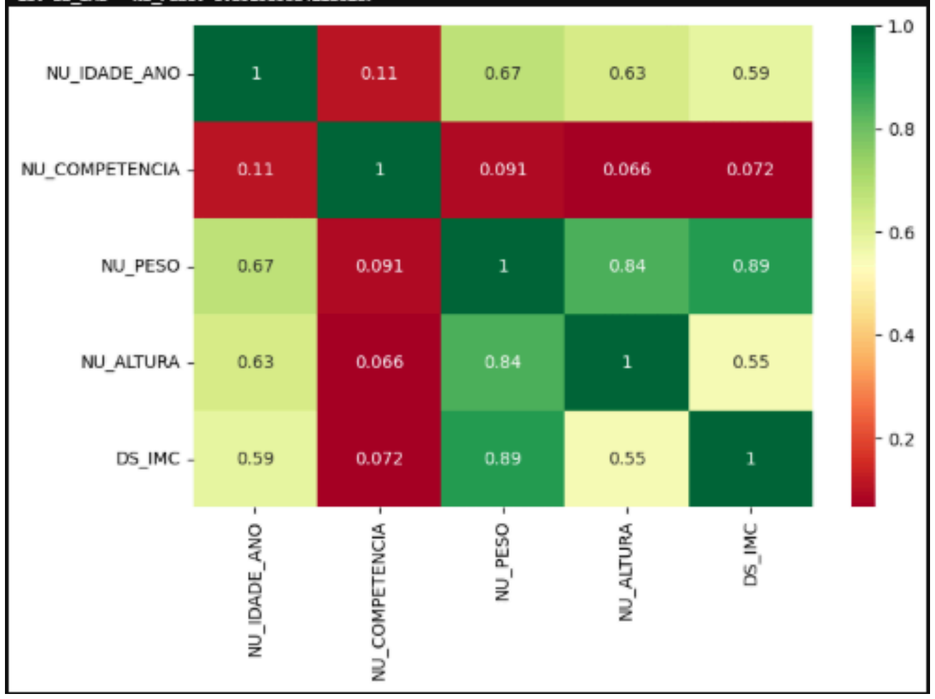
- Além disso, a variável DT_ACOMPANHAMENTO foi **convertida para o tipo datetime**, com tratamento de datas inválidas via errors='coerce', permitindo análises temporais consistentes e a extração de componentes como ano e mês.

- Foram aplicadas **regras fisiológicas** plausíveis para **garantir a integridade dos dados** antropométricos. **Removemos valores extremos de peso** (<1,5 kg ou >200 kg), altura (<40 cm ou >220 cm) e IMC (<10 ou >60), compatíveis com limites clínicos reais.

- A coluna de data de acompanhamento foi convertida corretamente para o formato datetime, assegurando consistência nas análises temporais.

Análise Exploratória

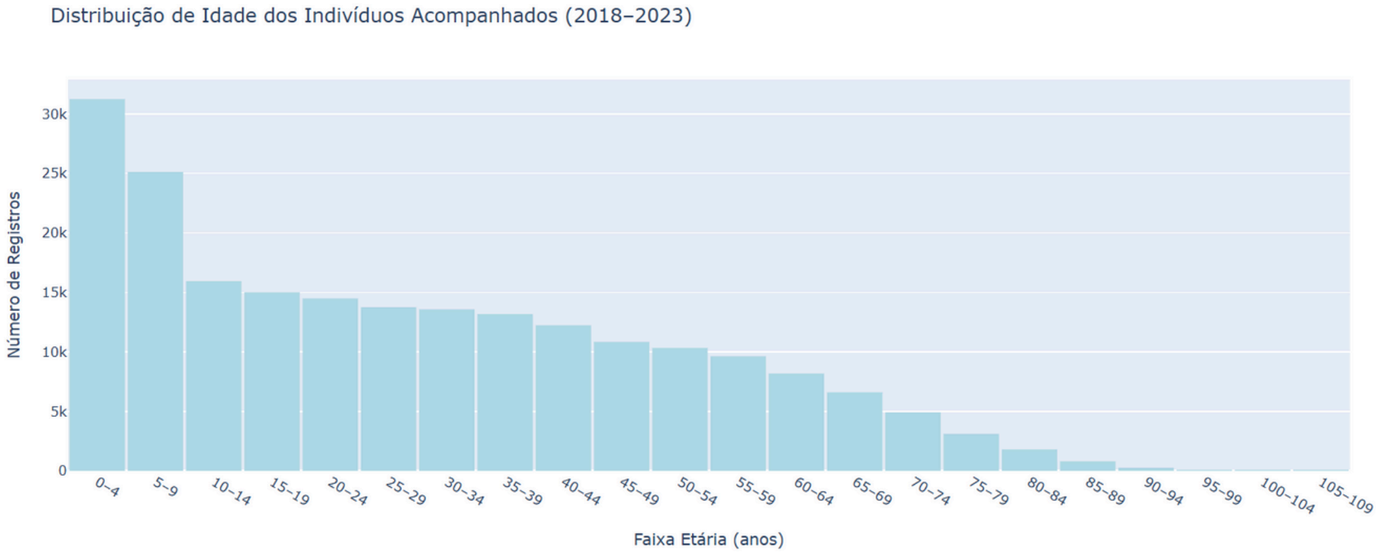
```
Maiores Correlações Positivas:
1: DS_IMC - NU_PESO: 0.8919095842185267
2: NU_PESO - NU_ALTURA: 0.8428693342329134
3: NU_PESO - NU_IDADE_ANO: 0.6700415182134181
4: NU_IDADE_ANO - NU_ALTURA: 0.6271030110358147
5: NU_IDADE_ANO - DS_IMC: 0.5855042895795757
6: NU_ALTURA - DS_IMC: 0.5497389297549022
7: NU_IDADE_ANO - NU_COMPETENCIA: 0.10903088784835477
8: NU_COMPETENCIA - NU_PESO: 0.09060547762161549
9: DS_IMC - NU_COMPETENCIA: 0.07166923887949712
10: NU_COMPETENCIA - NU_ALTURA: 0.06569807684936024

Maiores Correlações Negativas:
1: NU_COMPETENCIA - NU_ALTURA: 0.06569807684936024
2: DS_IMC - NU_COMPETENCIA: 0.07166923887949712
3: NU_COMPETENCIA - NU_PESO: 0.09060547762161549
4: NU_IDADE_ANO - NU_COMPETENCIA: 0.10903088784835477
5: NU_ALTURA - DS_IMC: 0.5497389297549022
6: NU_IDADE_ANO - DS_IMC: 0.5855042895795757
7: NU_IDADE_ANO - NU_ALTURA: 0.6271030110358147
8: NU_PESO - NU_IDADE_ANO: 0.6700415182134181
9: NU_PESO - NU_ALTURA: 0.8428693342329134
10: DS_IMC - NU_PESO: 0.8919095842185267
```



- As variáveis antropométricas apresentam **fortes correlações positivas**, como entre peso e IMC (0,89) e peso e altura (0,84), evidenciando coerência fisiológica nos dados;
- A idade também influencia diretamente peso, altura e IMC, reforçando seu papel como variável-chave nos modelos.

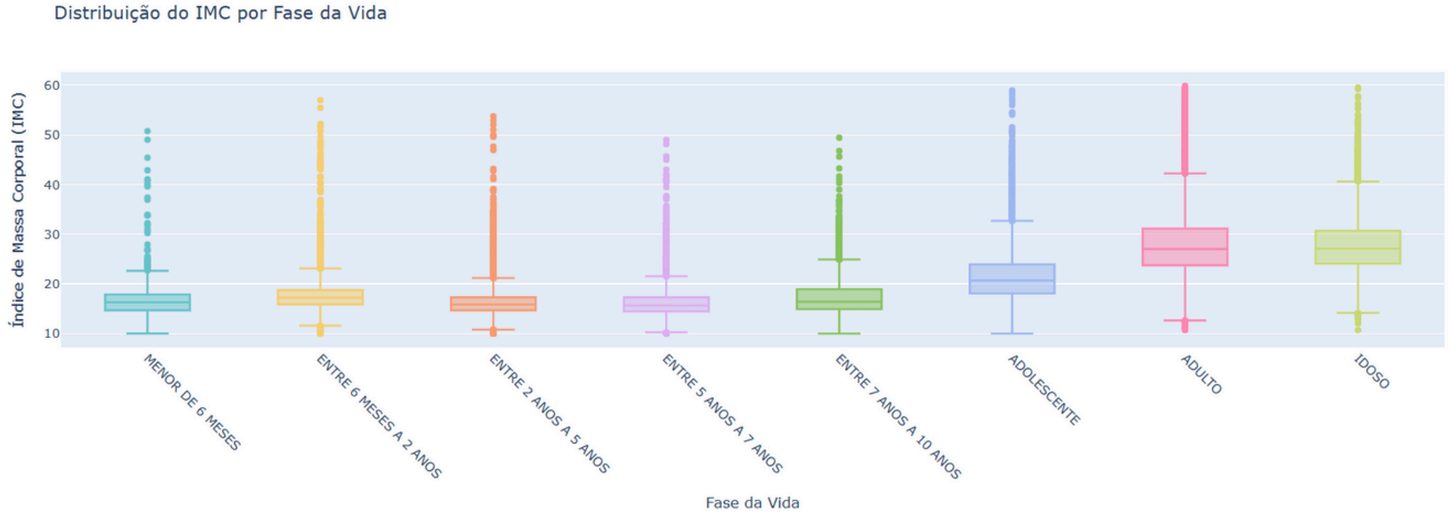
- As **correlações negativas observadas** com NU_COMPETENCIA (ano-mês do acompanhamento) **são fracas**, **indicando estabilidade temporal** nos padrões antropométricos da população analisada.

Análise Exploratória

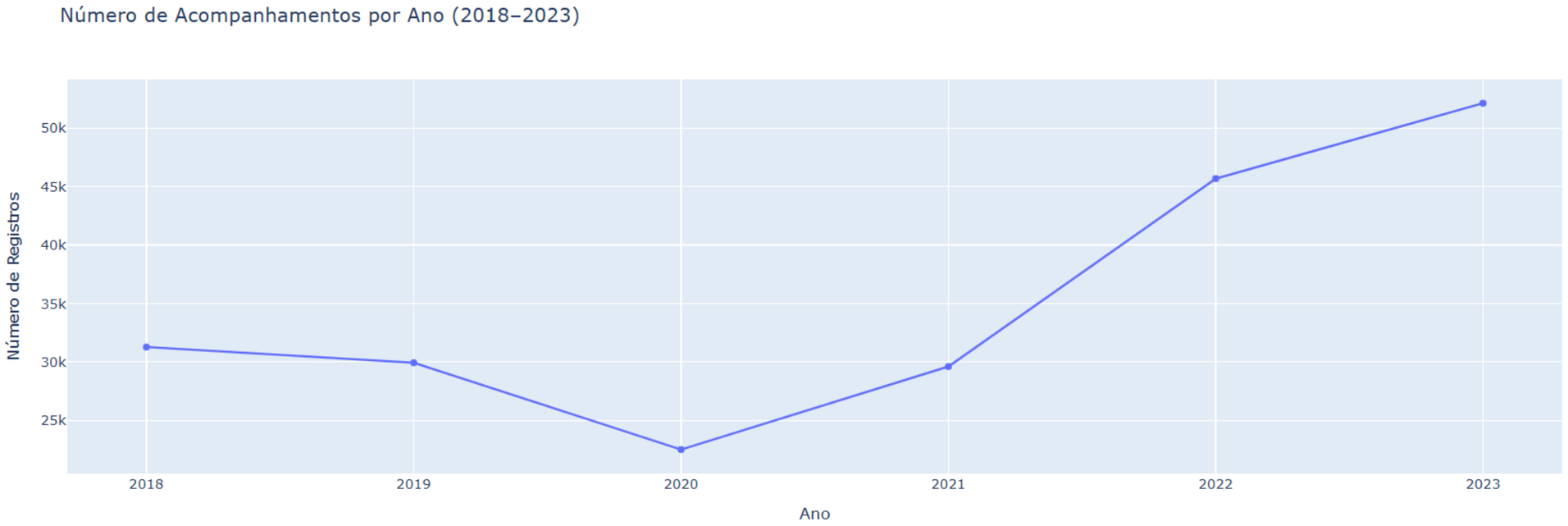


- O perfil dos dados indica um **foco maior nas faixas mais jovens**, seja por prioridade de programas, seja pela composição etária da população atendida.
- Políticas de saúde para **idosos** podem demandar mais atenção, visto o número reduzido de registros apesar da **alta prevalência de doenças crônicas nessa população**.
- A curva de distribuição pode ajudar no planejamento de recursos, priorizando regiões ou grupos etários com maior demanda.

- A distribuição do IMC nos mostra que o IMC **tende a aumentar com a idade**, especialmente a partir da adolescência.
- Maior variabilidade ocorre nas fases de transição hormonal (adolescência) e nas **fases de maior risco de doenças crônicas** (adultos e idosos).
- A presença de outliers elevados em todas as fases indica a **existência de casos de obesidade**, inclusive em **faixas etárias infantis**.



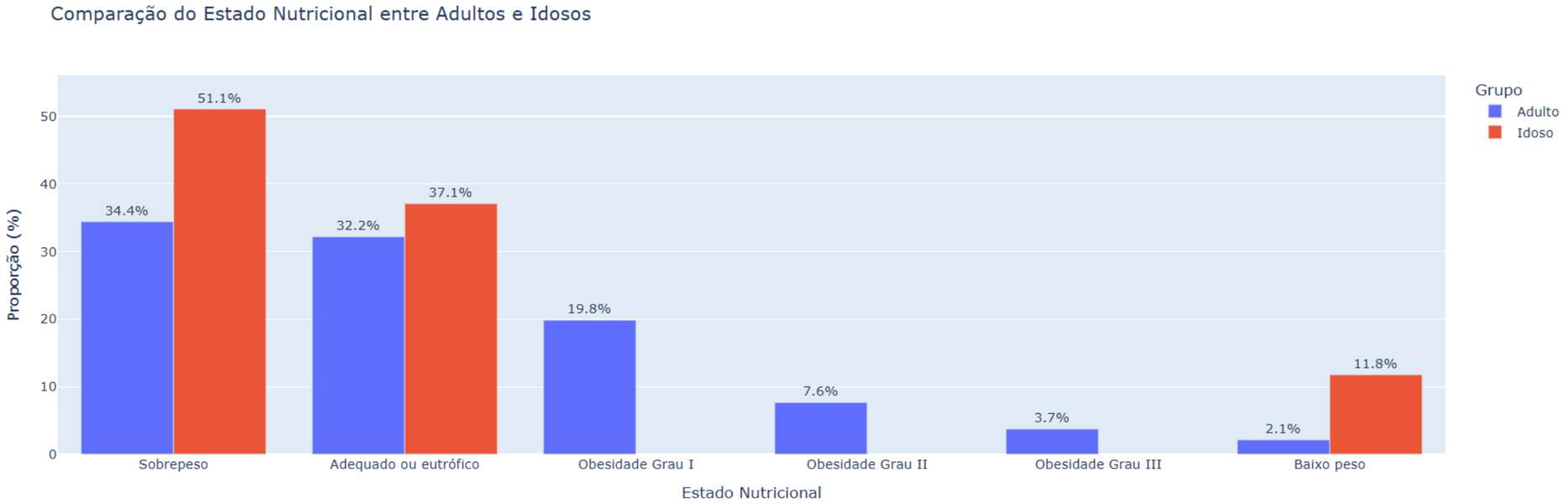
Análise Exploratória



	ANO	Atendimentos
0	2018	31280
1	2019	29936
2	2020	22516
3	2021	29607
4	2022	45679
5	2023	52120

- Houve **queda** no número de acompanhamentos **entre 2018 e 2020**, com o **menor valor em 2020**;
- **A partir de 2021**, observou-se uma **forte retomada**, culminando no **maior número de registros em 2023**;
- A **queda** em 2020 está associada à pandemia de **COVID-19**, enquanto a retomada reflete a **reorganização** e expansão dos serviços de saúde no **pós-pandemia**.

Análise Exploratória



- Os **idosos** apresentam **maior prevalência de sobrepeso e baixo peso**, o que sugere **dupla vulnerabilidade**: tanto à desnutrição quanto ao excesso de peso;
- Já os **adultos** concentram os casos de **obesidade nos graus mais elevados**, o que representa um alerta para **riscos metabólicos** e **cardiovasculares** nessa faixa etária;
- O padrão indica a necessidade de **estratégias nutricionais distintas** para cada grupo etário: **prevenção de obesidade** em adultos e **monitoramento** tanto de ganho quanto de perda de peso em idosos.

Execução do Modelo ARIMA

Previsão de OBESIDADE - ADULTOS

```
# Filtrar apenas adultos
df_adultos = df_consolidado[df_consolidado['DS_FASE_VIDA'] == 'ADULTO']

# Filtrar apenas quem tem estado nutricional disponível
df_adultos = df_adultos[df_adultos['CO_ESTADO_NUTRI_ADULTO'].notna()]

# Definir o que é considerado "obeso" para adultos
condicoes_obesidade = ['Sobrepeso', 'Obesidade Grau I', 'Obesidade Grau II', 'Obesidade Grau III']

# Série: percentual de obesos por ano
serie_obesidade_adultos = (
    df_adultos
    .groupby('ANO', group_keys=False)
    .apply(lambda x: (x['CO_ESTADO_NUTRI_ADULTO'].isin(condicoes_obesidade).sum() / len(x)) * 100)
    .reset_index(name='Percentual_Obesidade_Adultos')
)

serie_obesidade_adultos
```

	ANO	Percentual_Obesidade_Adultos
0	2018	61.900118
1	2019	62.653505
2	2020	66.026003
3	2021	67.353804
4	2022	66.317618
5	2023	67.753172

```
# Previsão para os próximos 5 anos
n_periods = 5
previsao, intervalo_conf = modelo.predict(n_periods=n_periods, return_conf_int=True)

# Criar DataFrame das previsões
anos_futuros = np.arange(2024, 2029)
df_previsoes = pd.DataFrame({
    'ANO': anos_futuros,
    'Previsao_Obesidade': previsao,
    'IC_Inferior': intervalo_conf[:, 0],
    'IC_Superior': intervalo_conf[:, 1]
})

df_previsoes
```

```
C:\Users\gsilv\miniforge3\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:837: ValueWarning:

No supported index is available. Prediction results will be given with an integer index beginning at `start`.
```

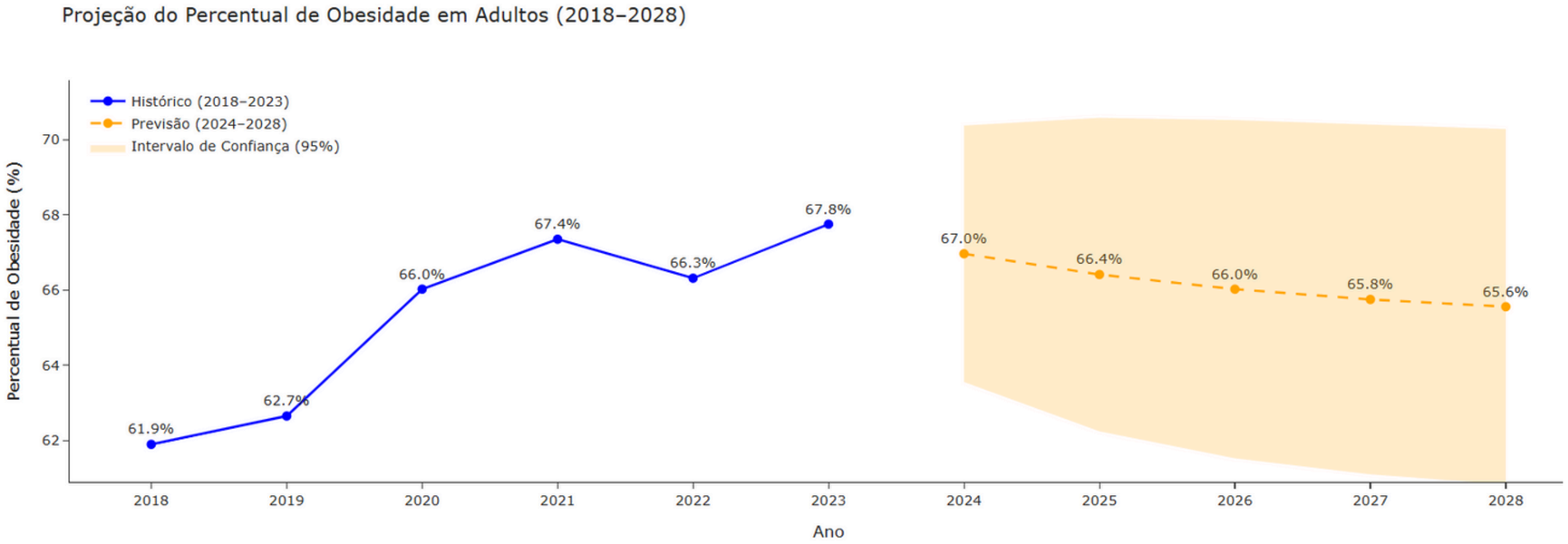
	ANO	Previsao_Obesidade	IC_Inferior	IC_Superior
6	2024	66.967026	63.558496	70.375557
7	2025	66.414198	62.247266	70.581130
8	2026	66.025442	61.530512	70.520372
9	2027	65.752063	61.103480	70.400647
10	2028	65.559820	60.837100	70.282540

- O código realiza uma análise temporal da obesidade entre adultos, permitindo **observar tendências** no percentual de obesos entre 2018 e 2023 — que, segundo os dados, tem mostrado uma tendência de **crescimento**;

- O código realiza uma **previsão da obesidade até 2028**, indicando que o **percentual** deve se manter relativamente **estável** com leve tendência de queda, porém com **intervalos de confiança amplos**, o que sugere incerteza sobre a precisão exata dos valores futuros.

Execução do Modelo ARIMA

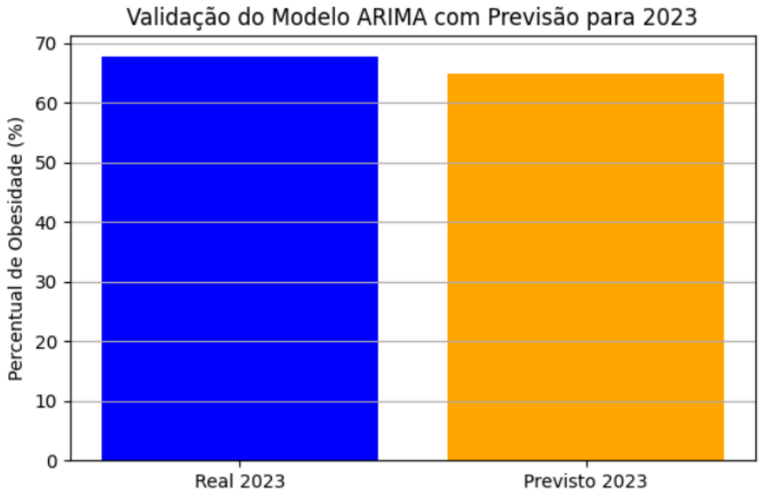
```python
import plotly.graph_objects as go

# Base para o gráfico
fig = go.Figure()

# Histórico
fig.add_trace(go.Scatter(
    x=serie.index,
    y=serie.values,
    mode='lines+markers+text',
    name='Histórico (2018–2023)',
    text=[f"{v:.1f}%" for v in serie.values],
    textposition="top center",
    line=dict(color='blue'),
    marker=dict(size=8)
))

# Previsão
fig.add_trace(go.Scatter(
    x=df_previsoes['ANO'],
    y=df_previsoes['Previsao_Obesidade'],
    mode='lines+markers+text',
    name='Previsão (2024–2028)',
    text=[f"{v:.1f}%" for v in df_previsoes['Previsao_Obesidade']],
    textposition="top center",
    line=dict(dash='dash', color='orange'),
    marker=dict(size=8)
))

# Banda de confiança
fig.add_trace(go.Scatter(
    x=pd.concat([df_previsoes['ANO'], df_previsoes['ANO'][::-1]]),
    y=pd.concat([df_previsoes['IC_Superior'], df_previsoes['IC_Inferior'][::-1]]),
    fill='toself',
    fillcolor='rgba(255,165,0,0.2)',
    line=dict(color='rgba(255,255,255,0)'),
    hoverinfo="skip",
    showlegend=True,
    name='Intervalo de Confiança (95%)'
))

# Linha divisória entre histórico e previsão
fig.add_vline(x=2023.5, line=dict(color="gray", dash="dot"))

# Layout
fig.update_layout(
    title='Projeção do Percentual de Obesidade em Adultos (2018–2028)',
    xaxis_title='Ano',
    yaxis_title='Percentual de Obesidade (%)',
    xaxis=dict(dtick=1),  # mostrar todos os anos
    yaxis=dict(range=[serie.min()-1, df_previsoes['IC_Superior'].max()+1]),
    legend=dict(x=0.01, y=0.99),
    template='simple_white'
)

fig.show()
```



- Apesar da alta histórica, o modelo sugere uma **estabilização ou ligeira redução** na obesidade adulta até 2028 — mas com incertezas consideráveis;
- O percentual de obesidade aumentou de 61,9% em 2018 para 67,8% em 2023;
- Destaque para o salto entre 2019 (62,7%) e 2020 (66,0%).

Execução do Modelo ARIMA

```python
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
plt.bar(['Real 2023', 'Previsto 2023'], [y_real_2023, previsao_2023], color=['blue', 'orange'])
plt.title('Validação do Modelo ARIMA com Previsão para 2023')
plt.ylabel('Percentual de Obesidade (%)')
plt.grid(True, axis='y')
plt.tight_layout()
plt.show()
```




```python
[ ] previsao_2023_array, intervalo_2023 = modelo_validacao.predict(n_periods=1, return_conf_int=True)

    # Acessar corretamente o valor previsto e intervalo
    previsao_2023 = previsao_2023_array.item()  # Extrai o valor float
    ic_inferior = intervalo_2023[0, 0]
    ic_superior = intervalo_2023[0, 1]

    print(f"Previsão para 2023: {previsao_2023:.2f}%")
    print(f"Intervalo de confiança: {ic_inferior:.2f}% a {ic_superior:.2f}%")
    print(f"Valor real em 2023: {y_real_2023:.2f}%")
```

```
C:\Users\gsilv\miniforge3\Lib\site-packages\statsmodels\tsa\base\tsa_model.py:837: ValueWarning:

No supported index is available. Prediction results will be given with an integer index beginning at `start`.

Previsão para 2023: 64.85%
Intervalo de confiança: 60.62% a 69.08%
Valor real em 2023: 67.75%
```

```python
[ ] mae = abs(y_real_2023 - previsao_2023)
    mape = (mae / y_real_2023) * 100

    print(f"MAE: {mae:.2f}%")
    print(f"MAPE: {mape:.2f}%")
```

```
MAE: 2.90%
MAPE: 4.28%
```

Apesar do modelo ARIMA ter apresentado uma previsão próxima do valor real, os resultados **ainda indicam limitações preocupantes na acurácia**:

- A previsão para 2023 (64,85%) ficou abaixo do valor real (67,75%), com uma **diferença de quase 3 pontos percentuais;**
- Mesmo **estando dentro do intervalo de confiança**, o modelo subestimou a obesidade, o que pode comprometer decisões baseadas nessas previsões — especialmente em áreas como saúde pública;
- O **erro percentual (MAPE) de 4,28%** pode parecer aceitável em alguns contextos, mas em projeções epidemiológicas, essa margem pode representar milhares de pessoas não contabilizadas adequadamente;
- O intervalo de confiança é relativamente amplo (de 60,62% a 69,08%), sugerindo incerteza considerável nas estimativas;

Em resumo, embora o modelo tenha tido um **desempenho estatisticamente aceitável**, ele demonstra fragilidades ao **capturar a real dinâmica da obesidade** e pode não ser suficientemente confiável para projeções de longo prazo ou para formulação de políticas públicas sem ajustes ou validações adicionais.

Extra: Rede Neural para Classificação de Estado Nutricional

```python
# Pré-processamento
X = pd.get_dummies(df[features], drop_first=True)
le = LabelEncoder()
y = le.fit_transform(df[target])
y = to_categorical(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```python
# Arquitetura da rede neural
inputs = Input(shape=(X_train.shape[1],), name="Input")
x = Dense(512, activation='relu', name="Dense_512")(inputs)
x = Dropout(0.5, name="Dropout_512")(x)
x = Dense(256, activation='relu', name="Dense_256")(x)
x = Dropout(0.4, name="Dropout_256")(x)
x = Dense(128, activation='relu', name="Dense_128")(x)
x = Dropout(0.3, name="Dropout_128")(x)
x = Dense(64, activation='relu', name="Dense_64")(x)
x = Dropout(0.2, name="Dropout_64")(x)
outputs = Dense(y.shape[1], activation='softmax', name="Output")(x)

model = Model(inputs=inputs, outputs=outputs, name="NN_Classifier")
model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```python
# Callbacks
callbacks = [
    EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True),
    ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3),
    ModelCheckpoint('best_model_classification.keras', save_best_only=True)
]
```

```python
classification_model(df=df_consolidado,
                     epochs=50,
                     features=['NU_IDADE_ANO', 'SG_SEXO', 'SG_UF', 'DS_RACA_COR', 'DS_ESCOLARIDADE', 'NU_PESO', 'NU_ALTURA', 'ANO'],
                     target='CO_ESTADO_NUTRI_ADULTO')
```

Para finalizar como extra modelamos uma rede neural para classificação do estado nutricional dos adultos, a decisão para incluir esse modelo vem da relevância dessa variável no dataset e será útil para entendermos quais variáveis se ajustam melhor à rede:

- Foi criada uma função que **codifica as variáveis categóricas** (*get_dummies*), **divide os dados em treino e teste**(*train_test_split*), **cria toda a arquitetura** faz o **ajuste** e **evaluation**;
- Arquitetura densa criada com a Functional API do Keras, com **camadas decrescentes (512 → 64)** e uso de **camadas de regularização *dropout*** para desativar aleatoriamente uma fração de neurônios **evitando *overfitting***;
- **Métricas** de **acurácia** e plot de **matriz de confusão**;

Callbacks Utilizados:

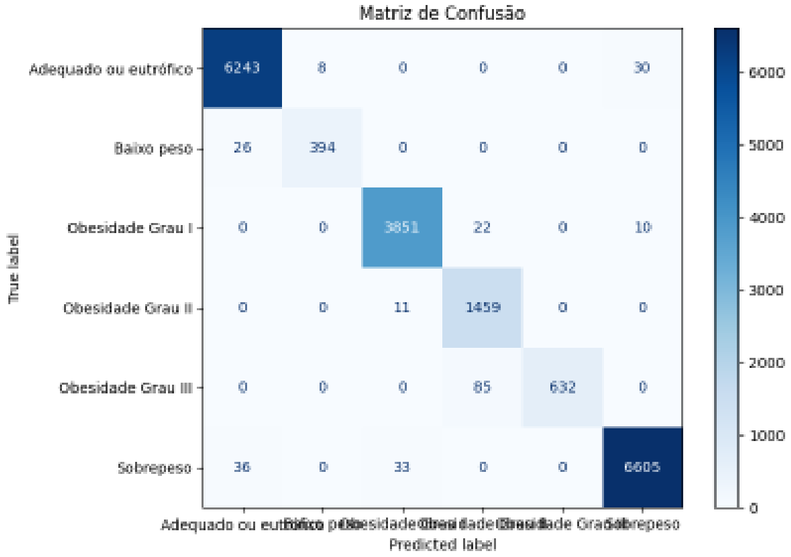
- *EarlyStopping*: **Interrompe o treinamento** automaticamente quando a **validação não melhora após várias épocas**. Isso evita overfitting e economiza tempo computacional;
- *ReduceLROnPlateau*: **Reduz a taxa de aprendizado** (learning rate) pela metade se a validação estagnar, permitindo que o otimizador faça ajustes mais finos e encontre um mínimo local mais estável;
- *ModelCheckpoint*: **Salva automaticamente o melhor modelo** (com menor val_loss) durante o treinamento, garantindo que não se perca uma boa configuração por overfitting nas épocas finais;

- Em sua chamada selecionamos quais variáveis features, target e quantidade de epochs desejados.

Extra: Rede Neural para Classificação de Estado Nutricional

```
Epoch 50/50
1945/1945 ━━━━━━━━━━━━━━━━━━━━ 3s 1ms/step - accuracy: 0.9828 - loss: 0.0456 - val_accuracy: 0.9879 - val_loss: 0.0337 - learning_rate: 6.2500e-05
608/608 ━━━━━━━━━━━━━━━━━━━━ 0s 590us/step
```

```
--- Classification Report ---
                      precision    recall  f1-score   support

Adequado ou eutrófico       0.99      0.99      0.99      6281
          Baixo peso        0.98      0.94      0.96       420
   Obesidade Grau I         0.99      0.99      0.99      3883
  Obesidade Grau II         0.93      0.99      0.96      1470
 Obesidade Grau III         1.00      0.88      0.94       717
         Sobrepeso          0.99      0.99      0.99      6674

            accuracy                           0.99     19445
           macro avg        0.98      0.96      0.97     19445
        weighted avg        0.99      0.99      0.99     19445
```



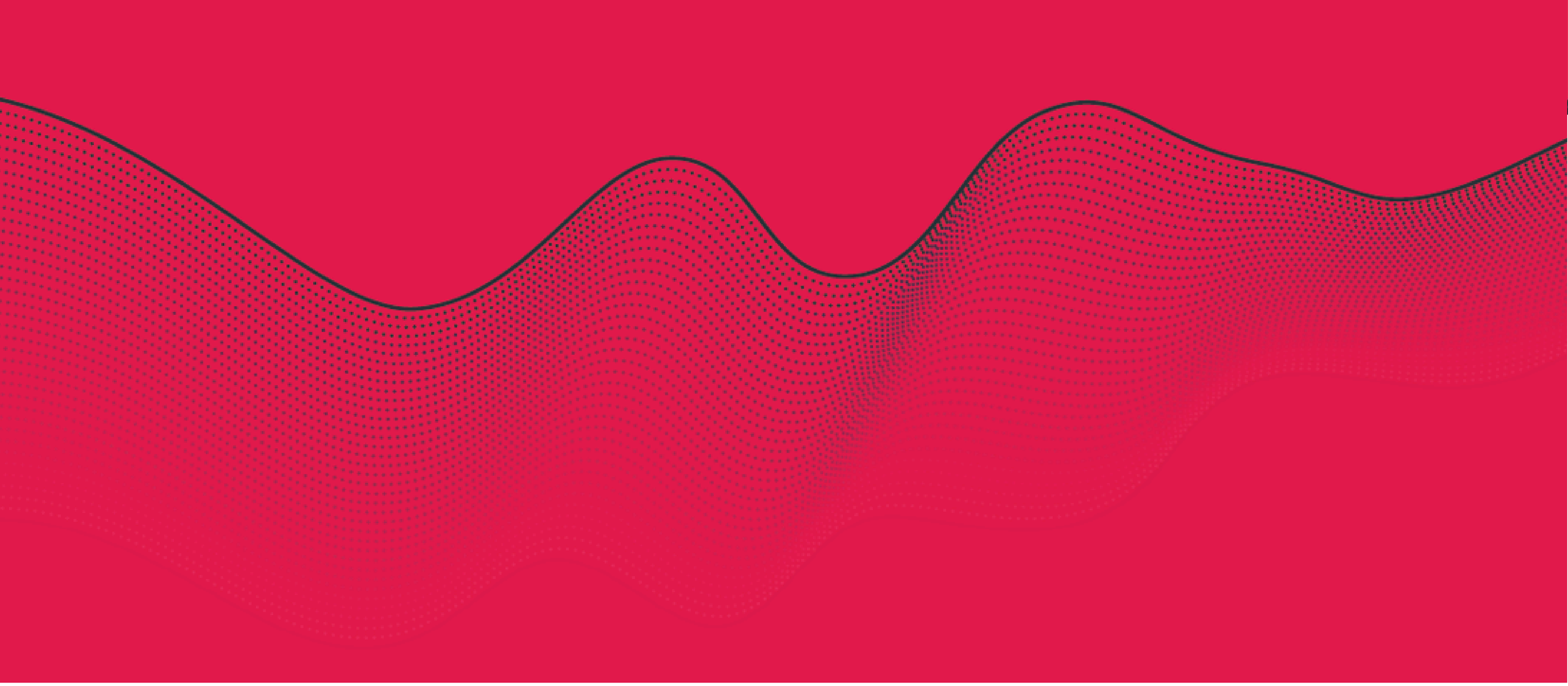
- Após 50 *epochs* de treinamento, a rede neural densa convergiu com **excelente desempenho**, alcançando **acurácia de 98,15%** e validação de 98,79%.

- O modelo apresentou **F1-score médio de 0,97**, confirmando sua **robustez na classificação multiclasse** do estado nutricional a partir de variáveis sociodemográficas e antropométricas.

- A matriz de confusão revela alta precisão na maioria das classes, com **destaque** para a correta identificação de indivíduos **eutróficos**, com **sobrepeso** e **obesidade grau I**.

Conclusões

- Este projeto utilizou dados retirados do **Sisvan (2018–2023)** para analisarmos de maneira aprofundada o estado nutricional da população atendida na Atenção Primária à Saúde e prevermos cenários futuros que possam gerar planos de execução de como melhorar esses indicadores;
- Para que isso fosse possível foi necessário **uma rigorosa etapa de pré-processamento e limpeza dos dados**, para então realizamos **diferentes tipos de análises exploratórias**;
- Através do modelo **ARIMA** foi possível entender como a **OBESIDADE** em **ADULTOS** se comportará nos anos seguintes ao último ano que temos registro, porém devido a limitações dos dados tivemos baixos níveis de acurácia do modelo;
- Para finalizar, fizemos uma **rede neural** para **classificar** o Estado Nutricional dos Adultos como um extra ao projeto e, mesmo com as limitações mencionadas, o modelo performou muito bem com valores **acima de 90% de acurácia**.



EaD

Universidade Presbiteriana
Mackenzie