Relatório de Resultados

TÓPICOS EM INTELIGÊNCIA ARTIFICIAL II

Universidade Federal Fluminense

Camila Ferreira

1. Datasets Escolhidos:

Dataset 1:

Estimation of Obesity Levels Dataset

Fonte: UCI Machine Learning Repository - Dataset 544

Descrição

Este dataset foi criado com o objetivo de prever os níveis de obesidade com base em hábitos alimentares e condições físicas dos indivíduos. Ele foi coletado por meio de questionários preenchidos por pessoas no México, Peru e Colômbia. A base contém variáveis relacionadas a estilo de vida, como frequência de exercícios, consumo de alimentos e histórico de peso.

Informações do Dataset

- **Tamanho**: 2111 instâncias
- **Atributos**: 17 atributos

Atributo Alvo (Target)

- NObeyesdad: Nível de obesidade do indivíduo
  - Normal_Weight
  - Overweight_Level_I
  - Overweight_Level_II
  - Obesity_Type_I
  - Insufficient_Weight
  - Obesity_Type_II
  - Obesity_Type_III

Atributo	Tipo	Descrição
Gender	Categórica	Gênero do indivíduo
Age	Numérica	Idade
Height	Numérica	Altura em metros
Weight	Numérica	Peso em kg
family_history_with_overweight	Booleana	Histórico familiar de sobrepeso
FAVC	Booleana	Consome alimentos calóricos com frequência?
FCVC	Numérica	Frequência de consumo de vegetais
NCP	Numérica	Número de refeições principais por dia
CAEC	Categórica	Consumo de alimentos entre as refeições
SMOKE	Booleana	Fuma

Atributo	Tipo	Descrição
CH2O	Numérica	Consumo diário de água (litros)
SCC	Booleana	Monitora ingestão calórica?
FAF	Numérica	Frequência de atividade física (horas por semana)
TUE	Numérica	Tempo usando tecnologia por dia (horas)
CALC	Categórica	Frequência de consumo de álcool
MTRANS	Categórica	Meio de transporte mais utilizado

Dataset 2:

Contraceptive Method Choice Dataset

Fonte: UCI Machine Learning Repository - Dataset 30

Descrição

Este dataset foi desenvolvido com o objetivo de prever o método contraceptivo utilizado por mulheres com base em informações socioeconômicas e demográficas. A base é derivada de um estudo realizado na Indonésia e é composta por dados coletados de mulheres casadas que participavam do programa nacional de planejamento familiar. O foco principal é auxiliar na análise de padrões que influenciam a escolha de métodos contraceptivos.

Informações do Dataset

- **Tamanho**: 1473 instâncias
- **Atributos**: 9 atributos

Atributo Alvo (Target)

- **Contraceptive Method Used**: Método contraceptivo utilizado pela mulher
  - 1: Nenhum
  - 2: Curto prazo (pílulas, preservativos, etc.)
  - 3: Longo prazo (DIU, esterilização, etc.)

Atributo	Tipo	Descrição
Wife_age	Numérica	Idade da esposa (em anos)
Wife_education	Categórica	Grau de escolaridade da esposa (1=baixo, 4=alto)
Husband_education	Categórica	Grau de escolaridade do marido (1=baixo, 4=alto)
Num_children	Numérica	Número de filhos vivos
Wife_religion	Booleana	Religião da esposa (0=Islâmica, 1=Outra)

Atributo	Tipo	Descrição
Wife_working	Booleana	A esposa trabalha? (0=Não, 1=Sim)
Husband_occupation	Categórica	Ocupação do marido (valores de 1 a 4)
Standard_of_living	Categórica	Nível de vida (1=baixo, 4=alto)
Media_exposure	Booleana	Exposição à mídia (0=Não exposta, 1=Exposta)

Dataset 3:

Heart Disease Dataset

Fonte: UCI Machine Learning Repository - Dataset 45

Descrição

Este dataset foi criado com o objetivo de prever a presença de doenças cardíacas em pacientes com base em diversas características clínicas e demográficas. A base contém dados coletados de quatro bancos hospitalares diferentes, sendo a versão mais utilizada aquela que consolida as informações em um único conjunto limpo e padronizado. Os atributos incluem variáveis como idade, sexo, pressão arterial, nível de colesterol e resultados de exames cardíacos.

Informações do Dataset

- **Tamanho**: 303 instâncias (na versão Cleveland, a mais usada)
- **Atributos**: 13 atributos

Atributo Alvo (Target)

- **Grau de doença cardíaca presente no indivíduo** : valores de 0 a 4, onde 0 significa ausência da doença

Atributo	Tipo	Descrição
age	Numérica	Idade do paciente
sex	Binária	Sexo (1 = masculino; 0 = feminino)
cp	Categórica	Tipo de dor no peito (0 a 3)
trestbps	Numérica	Pressão arterial em repouso (mm Hg)
chol	Numérica	Nível sérico de colesterol (mg/dl)
fbs	Binária	Glicemia em jejum > 120 mg/dl (1 = verdadeiro; 0 = falso)
restecg	Categórica	Resultados do eletrocardiograma em repouso
thalach	Numérica	Frequência cardíaca máxima atingida
exang	Binária	Angina induzida por exercício (1 = sim; 0 = não)

Atributo	Tipo	Descrição
oldpeak	Numérica	Depressão do segmento ST induzida por exercício em relação ao repouso
slope	Categórica	Inclinação do segmento ST no esforço máximo
ca	Numérica	Número de vasos principais coloridos por fluoroscopia (0 a 3)
thal	Categórica	Resultado do teste de tálio (3 = normal; 6 = defeito fixo; 7 = reversível)

2. Tratamento de dados

As etapas realizadas foram:

- **Identificação de tipos de dados**: As colunas foram separadas entre categóricas e numéricas. As binárias, (com valores *"yes"/"no"*) foram tratadas como categóricas.
- **Codificação de variáveis categóricas**: Foi aplicado o *OneHotEncoder* para transformar variáveis categóricas em um formato numérico apropriado.
- **Normalização de dados numéricos**: Utilizou-se *StandardScaler* para padronizar os atributos numéricos, garantindo que todos estivessem na mesma escala.
- **Combinação dos dados tratados**: Os dados normalizados e codificados foram combinados em uma única matriz, que então foi convertida em um novo DataFrame com nomes de colunas apropriados.

3. Implementação da Rede Neural

A arquitetura do modelo baseia-se em uma *Multi-Layer Perceptron (MLP)* implementada com *PyTorch*. A classe *MLP* define uma rede neural totalmente conectada com múltiplas camadas ocultas, função de ativação customizável e regularização via *Dropout*.

A classe *MLPClassifierTorch* encapsula a MLP dentro de uma interface compatível com *scikit-learn*, permitindo uso com *Pipeline* e integração com métodos de validação e otimização como *RandomizedSearchCV*.

3.1. Otimização de Hiperparâmetros

A seleção dos hiperparâmetros é feita por meio de *RandomizedSearchCV*, com um número váriável de iterações para cada dataset e validação cruzada *StratifiedKFold* . O espaço de busca abrange:

- *hidden_sizes*: tamanhos das camadas ocultas (ex: (64,), (128,), (128, 64), (256, 128)).
- *dropout_rate*: taxa de dropout (amostrada de uma distribuição contínua entre 0.1 e 0.5).
- *learning_rate*: taxa de aprendizado (amostrada entre 0.0001 e 0.01).
- *activation_fn*: função de ativação (*ReLU*, *LeakyReLU*, *Tanh*).
- *weight_decay*: regularização L2.
- *max_epochs*, *early_stopping*, *patience*, *verbose*.

3.2 Avaliação do Modelo

O melhor modelo identificado pela busca aleatória é avaliado nos conjuntos de treino e teste. As principais métricas utilizadas são:

- **Acurácia** global
- **Relatório de classificação** com precisão, recall e F1-score por classe
- **Matriz de confusão**, visualizada com *seaborn*

A matriz de confusão permite identificar padrões de erro entre classes reais e previstas, sendo útil especialmente em problemas multiclasse.

Exemplo de geração das métricas:

```
from sklearn.metrics import classification_report, confusion_matrix

print(classification_report(y_test, y_pred, target_names=target_names))

sns.heatmap(confusion_matrix(y_test, y_pred),
            annot=True, fmt='d', cmap='Blues',
            xticklabels=target_names, yticklabels=target_names)
```

3.3 Balanceamento de Classes com SMOTE

Durante a análise dos dados, foi identificado que o *Dataset 3* e o *Dataset 2* apresentava um desbalanceamento significativo entre as classes da variável alvo. Para resolver esse problema, utilizamos o *SMOTE* (Synthetic Minority Over-sampling Technique, uma técnica de oversampling que cria novas instâncias sintéticas da(s) classe(s) minoritária(s).

Por que apenas no Dataset 2 e 3?

O Dataset 1 apresentava distribuição relativamente equilibrada entre as classes, ou um desequilíbrio que não impactava negativamente o desempenho do modelo. Já os Datasets 2 e 3 apresentava um forte viés, com poucas instâncias representando uma das classes, o que exigiu intervenção para evitar que o modelo aprendesse apenas padrões da classe majoritária.

Etapas Realizadas no SMOTE

1. **Limpeza dos dados**:
    - Remoção de instâncias com valores ausentes utilizando *.dropna()*.
2. **Separação da variável alvo (*y*) correspondente às instâncias válidas.**
3. **Aplicação do SMOTE**:
    - Utilizado com *random_state=42* para reprodutibilidade.
    - Gerou amostras sintéticas para balancear as classes.

```python
from imblearn.over_sampling import SMOTE

X_cleaned_df = X_processed_df.dropna()
y_cleaned = y[X_processed_df.notna().all(axis=1)]

smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_cleaned_df, y_cleaned)
```

---

4. Resultados:

Dataset 1:

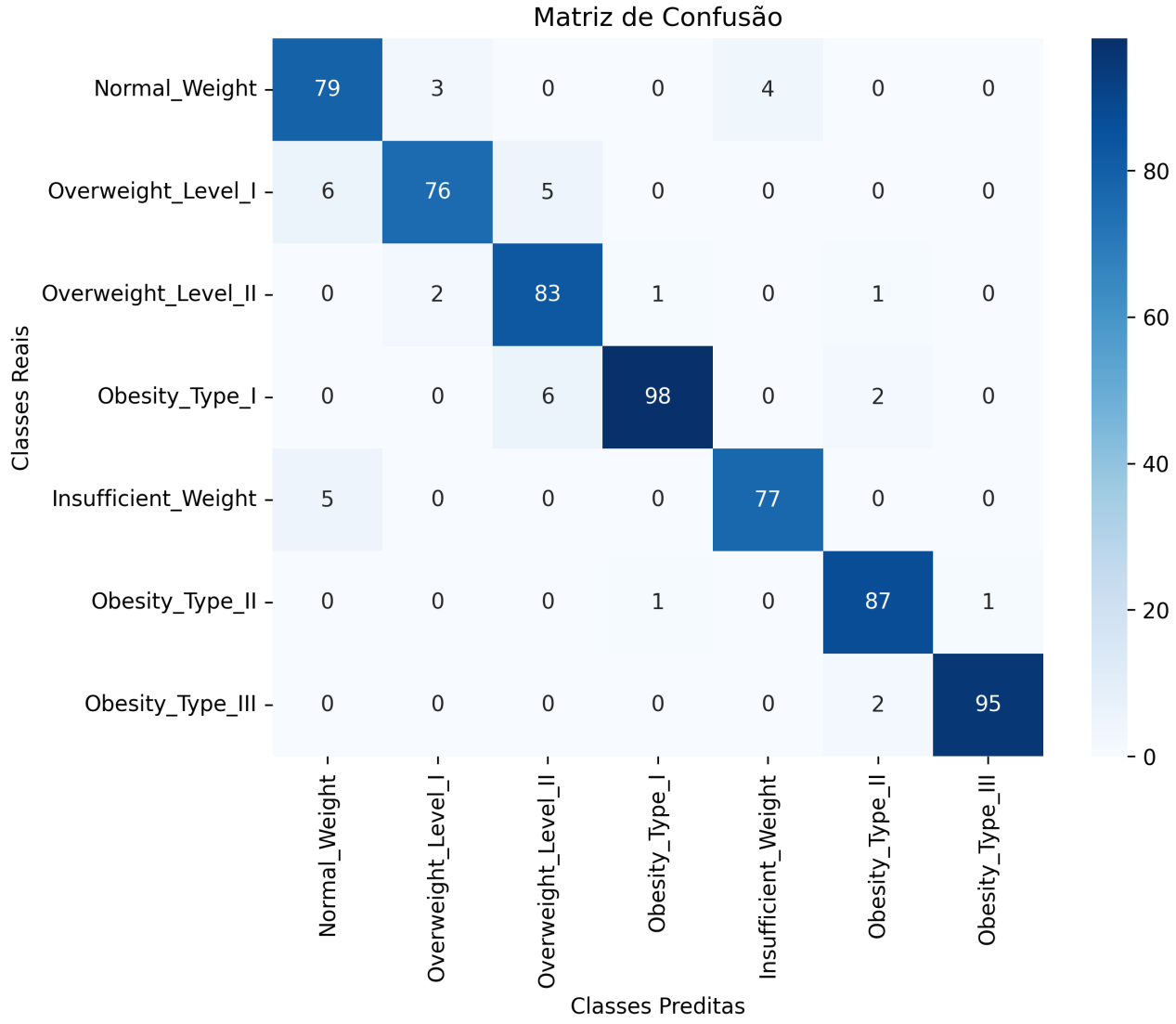
1. **Métricas por classe :**

```
Melhores hiperparâmetros encontrados:
{'mlp__activation_fn': <class 'torch.nn.modules.activation.Tanh'>, 'mlp__dropout_rate': np.float64(0.14998745790900145), 'mlp__early_stopp
ing': True, 'mlp__hidden_sizes': (64, 32), 'mlp__learning_rate': np.float64(0.008761761457749352), 'mlp__max_epochs': 129, 'mlp__patienc
e': 10, 'mlp__verbose': False, 'mlp__weight_decay': np.float64(0.001438668179219408)}
Acurácia no conjunto de teste: 93.85%
Relatório de Classificação:
                     precision    recall  f1-score   support

      Normal_Weight       0.88      0.92      0.90        86
 Overweight_Level_I       0.94      0.87      0.90        87
Overweight_Level_II       0.88      0.95      0.92        87
     Obesity_Type_I       0.98      0.92      0.95       106
Insufficient_Weight       0.95      0.94      0.94        82
    Obesity_Type_II       0.95      0.98      0.96        89
   Obesity_Type_III       0.99      0.98      0.98        97

           accuracy                           0.94       634
          macro avg       0.94      0.94      0.94       634
       weighted avg       0.94      0.94      0.94       634
```

2. **Matriz Confusão :**



Dataset 2:

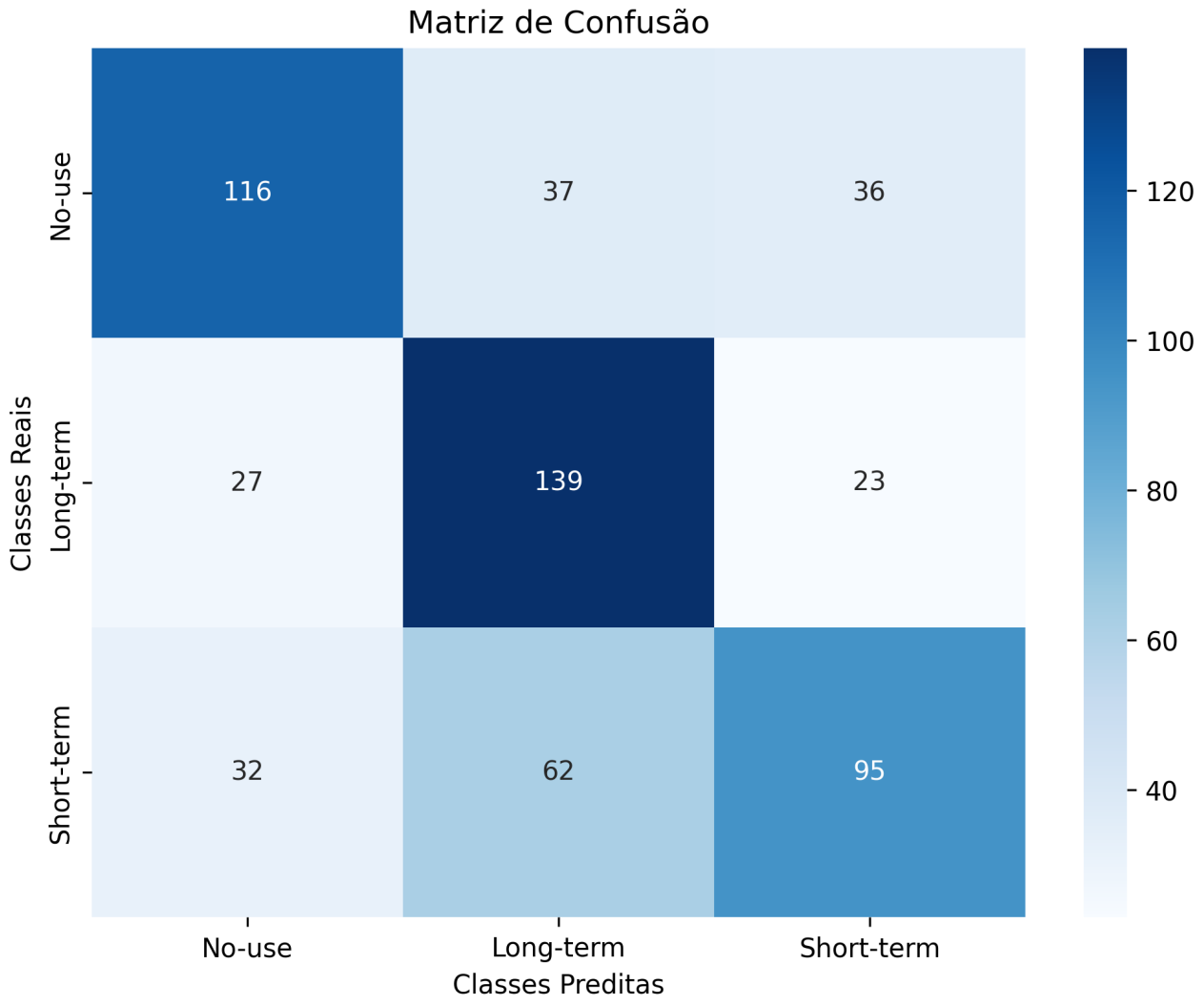
1. **Métricas por classe :**

```
Melhores hiperparâmetros encontrados:
{'mlp__activation_fn': <class 'torch.nn.modules.activation.ReLU'>, 'mlp__dropout_rate': np.float64(0.5357302950938588), 'mlp__early_stoppi
ng': True, 'mlp__hidden_sizes': (128, 64), 'mlp__learning_rate': np.float64(0.002284404372168336), 'mlp__max_epochs': 157, 'mlp__patienc
e': 10, 'mlp__verbose': False, 'mlp__weight_decay': np.float64(0.008084401551640625)}
Acurácia no conjunto de teste: 61.73%
Relatório de Classificação:
              precision    recall  f1-score   support

      No-use       0.66      0.61      0.64       189
   Long-term       0.58      0.74      0.65       189
  Short-term       0.62      0.50      0.55       189

    accuracy                           0.62       567
   macro avg       0.62      0.62      0.61       567
weighted avg       0.62      0.62      0.61       567
```

2. **Matriz Confusão :**



Dataset 3:

1. **Métricas por classe :**

```
Melhores hiperparâmetros encontrados:
{'mlp__activation_fn': <class 'torch.nn.modules.activation.ReLU'>, 'mlp__dropout_rate': np.float64(0.1376731280030641), 'mlp__early_stoppi
ng': True, 'mlp__hidden_sizes': (256, 128), 'mlp__learning_rate': np.float64(0.003950977286019253), 'mlp__max_epochs': 190, 'mlp__patienc
e': 10, 'mlp__verbose': False, 'mlp__weight_decay': np.float64(0.0031792200515627766)}
Acurácia no conjunto de teste: 85.54%
Relatório de Classificação:
                    precision    recall  f1-score   support

        No disease       0.95      0.76      0.84        49
       Mild disease       0.69      0.85      0.77        48
   Moderate disease       0.89      0.81      0.85        48
Significant disease       0.88      0.86      0.87        49
     Severe disease       0.92      1.00      0.96        48

           accuracy                           0.86       242
          macro avg       0.87      0.86      0.86       242
       weighted avg       0.87      0.86      0.86       242
```

2. **Matriz Confusão :**