Projeto de Data Science grupo 10

1st Henrique Zan Grande
Departamento de informática
PUC-PR
Curitiba, Brasil
henrique.grande@pucpr.edu.br

2nd João Gabriel Pitol
Departamento de informática
PUC-PR
Curitiba, Brasil
joao.pitol@pucpr.edu.br

3rd Lucas Braga Schuck
Departamento de informática
PUC-PR
Curitiba, Brasil
lucas.shuck.pucpr.edu.br

4th Rafael Vargas Serenato
Departamento de informática
PUC-PR
Curitiba, Brasil
rafael.serenato@pucpr.edu.br

***Abstract*—Este trabalho tem como objetivo a análise de dados clínicos, comportamentais e sociodemográficos para o diagnóstico da Doença de Alzheimer. A base de dados utilizada foi obtida da plataforma Kaggle, contendo informações de 2.149 pacientes idosos. O estudo faz uma analise geral dos dados e suas relações e em seguida aplica técnicas de machine learning para criar modelos preditivos.**

I. Introdução

Este estudo visa analisar um conjunto de dados clínicos, comportamentais e sociodemográficos de pacientes idosos, com foco no diagnóstico da Doença de Alzheimer. Através da aplicação de algoritmos de machine learning, buscamos identificar e compreender padrões que possam contribuir para a antecipação do diagnóstico, utilizando técnicas para melhorar a performance dos modelos.

II. Descrição da Base de Dados

A. Fonte

A base de dados utilizada neste projeto é a "Alzheimer's Disease Dataset", obtida a partir da plataforma Kaggle. Ela contém informações clínicas, comportamentais e sociodemográficas de 2.149 pacientes idosos.

B. Estrutura da Base de Dados

O conjunto de dados contém 35 colunas (variáveis) e 2.149 linhas (instâncias). Após uma limpeza inicial, foram removidas as colunas não informativas, como PatientID e DoctorInCharge, restando 33 colunas com informações relevantes.

A variável target, "Diagnosis", é binária, indicando se o paciente apresenta (positivo) ou não (negativo) o diagnóstico da doença.

C. Desbalanceamento da Base

Foi identificado um desbalanceamento significativo na variável target, com a classe "negative" representando aproximadamente 65% das amostras e a classe "positive" representando 35%.

III. Análise Exploratória dos Dados

A. Análise Univariada

A análise univariada teve como objetivo compreender a distribuição das variáveis. Para isso, foram utilizados gráficos de barras e histogramas, evidenciando as distribuições das variáveis. Além disso, foi realizada uma análise estatística descritiva, revelando médias, desvios padrão e limites para variáveis numéricas.

Essa etapa nos fez entender o comportamento das variáveis individualmente e um dos pontos observados foi que as variáveis numéricas apresentaram distribuições bastante uniformes, sem assimetrias relevantes, provavelmente pois o Dataset se tratava de uma população composta por pessoas idosas com algum tipo de declínio mental e tendo condições de saúde muito semelhantes, já as variáveis categóricas em sua maioria apresentaram um grande desbalanceamento entre as classes, principalmente quando são relacionadas a presença de sintomas.

B. Análise Multivariada

A análise multivariada teve como objetivo entender as relações entre as variáveis e a variável target. Visualizações foram realizadas para evidenciar grupos de sintomas ou comportamentos com correlação com o desenvolvimento da doença.

Esta etapa nos ajudou a entender que variáveis relacionadas ao comportamento e capacidade funcional demonstraram relações fortes com o diagnóstico nos mostrando o caminho em que seguimos nas análises.

IV. Pré-processamento

A. Padronização dos Dados

Para obter os melhores resultados dos modelos selecionados (KNN e SVM), foi realizada a padronização dos dados. A variável target "Diagnosis" foi transformada em valores binários (0 e 1) com LabelEncoder, e as variáveis preditoras foram separadas em numéricas, ordinais e nominais. Para as variáveis numéricas, utilizou-se o StandardScaler para padronizá-las.

A variável ordinal "EducationLevel" foi codificada com OrdinalEncoder, enquanto as variáveis nominais foram transformadas com OneHotEncoder, removendo a coluna da primeira categoria para evitar multicolinearidade.

B. Divisão em folds

Foi utilizada a validação cruzada estratificada com 10 folds (Stratified K-Fold Cross-Validation), garantindo que a proporção de classes fosse mantida em cada subdivisão, e cada subdivisão permanecesse igual para cada modelo usando o mesmo RandomState=42.

C. Balanceamento das Classes

O desbalanceamento da variável alvo pode induzir o modelo a favorecer a classe majoritária, prejudicando o reconhecimento das minoritárias. Para contornar esse problema aplicaram-se as seguintes técnicas:

- **Random Over Sampling (ROS)**: Duplica aleatoriamente instâncias da classe minoritária, mantendo todas as instâncias originais, mas podendo causar overfitting.
- **SMOTE** (Synthetic Minority Over-sampling Technique): Gera novas amostras sintéticas,em uma zona segura, interpolando entre vizinhos minoritários, aumentando a diversidade, mas sensível a ruído.
- **Random Under Sampling (RUS)**: Remove aleatoriamente instâncias da classe majoritária, reduzindo o tamanho do conjunto e acelerando o treinamento, porém pode descartar exemplos informativos.
- **NearMiss (v1)**: Seleciona exemplos majoritários mais próximos das minoritárias com base na distância média.

V. Modelos de Machine Learning Utilizados

A. K-Nearest Neighbors (KNN)

O modelo KNN é baseado na proximidade de instâncias, foi configurado apenas utilizando o parâmetro $n\_neighbors = 5$ e teve os piores resultados dentre os 3 modelos.

B. Random Forest (RF)

O algoritmo Random Forest é baseado em ensembles de árvores de decisão, como desde o inicio dos testes este modelo teve os melhores resultados, aplicamos modificações nos hyperparametros para melhorar ainda mais a performance e os melhores encontrados foram: n_estimators=200, max_depth=10, min_samples_split=5, min_samples_leaf=2, max_features='sqrt', class_weight='balanced' e radom_state=42.

C. Support Vector Machine (SVM)

O modelo SVM foi utilizado com o kernel radial (rbf) e configurado com o randomstate=42, teve resultados intermediários considerando os 3 modelos.

VI. Protocolo de Validação e Teste Estatístico

A. Protocolo

Para avaliar o desempenho dos modelos de forma consistente, foi utilizado o protocolo de validação cruzada estratificada com 10 folds. Essa abordagem garantiu que cada subdivisão dos dados preserve a proporção original das classes e usando o mesmo valor de randomstate=42 pudemos assegurar que todos os modelos fossem testados com as mesmas partições, permitindo comparações justas e reproduzíveis entre eles.

Cada um dos 3 modelos foi treinado com os 4 tipos de balanceamento e testados/validados em conjutos de testes que não foram balanceados (seguindo a proporção padrão da variavél target presente no dataset original).

Como resultado foram obtidas as métricas citadas abaixo:

B. Métricas de Avaliação

- **Acurácia**: Para medir a poporção de predições corretas sendo positivas ou negativas.
- **Precisão**: Para medir a quantos exemplos classificados como positivo foram classificados corretamente.
- **Recall**: Para medir a capacidade do modelo de identificar todos os casos positivos reais.
- **F1-score**: média harmônica entre precisão e recall, encontra um equilibrio na identificação de falsos positivos e negativos.

C. Teste de Friedman e Análise Post-hoc

Para verificar se havia diferença real de desempenho entre os modelos, usamos o teste de Friedman, um método não paramétrico indicado para comparar várias técnicas em amostras relacionadas, sem exigir que os dados tenham uma distribuição normal. Adotamos um nível de significância de 5% ($p < 0{,}05$) para decidir se rejeitávamos a hipótese de que todos os classificadores têm desempenho igual.

Como o teste apontou diferenças significativas, seguimos para uma análise post-hoc com a biblioteca autorank. Essa etapa produziu um ranking, onde cada modelo recebe uma posição conforme sua média de colocações, mostrando claramente quais modelos não têm diferença estatística relevante.

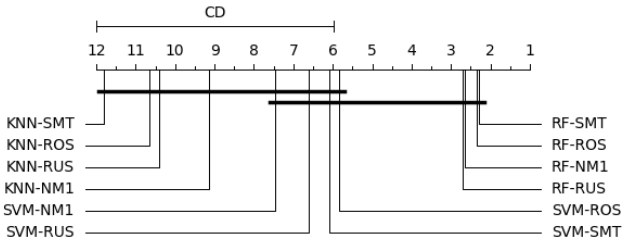
VII. Resultados Obtidos

Os resultados das médias das métricas, calculadas para cada fold de cada modelo, estão detalhadamente apresentadas na Tabela I. Essas métricas nos mostram uma visão geral do desempenho de cada modelo durante a validação. Além disso, para uma análise mais visual das diferenças estatisticas de desempenho entre os modelos, um gráfico de teste estatístico foi gerado. Este gráfico, ilustra se existem diferenças significativas entre os modelos e está disponível na Imagem 1, fazendo uma comparação clara e entre as abordagens testadas.

TABLE I
MÉTRICAS DOS MODELOS E TÉCNICAS DE BALANCEAMENTO

Modelo - Balanceamento	Acurácia	Precision	Recall	F1-score
KNN - RandomOver	0.6994	0.5612	0.6882	0.6174
KNN - SMOTE	0.6603	0.5129	0.7895	0.6218
KNN - RandomUnder	0.7087	0.5719	0.7158	0.6352
KNN - NearMiss-1	0.7446	0.6544	0.5908	0.6203
RandomForest - RandomOver	0.9395	0.9402	0.8855	0.9118
RandomForest - SMOTE	0.9427	0.9481	0.8868	0.9160
RandomForest - RandomUnder	0.9418	0.9208	0.9145	0.9174
RandomForest - NearMiss-1	0.9362	0.9016	0.9211	0.9109
SVM - RandomOver	0.8404	0.7671	0.7895	0.7777
SVM - SMOTE	0.8353	0.7750	0.7540	0.7639
SVM - RandomUnder	0.8311	0.7311	0.8276	0.7758
SVM - NearMiss-1	0.8125	0.7082	0.8053	0.7526

Fig. 1. Teste de Nemenyi



A. Conclusões dos Resultados

O gráfico demonstra que o os melhores modelos, todos os Random Forest, não apresentam diferenças significativas quando comparados com os modelos de SVM mas apresentam uma significativa diferença quando comparados com os todos modelos KNN.

Já os modelos SVM não apresentam diferenças significativas com nenhum modelo testado, evidenciando bem a sua posição ”central” no ranking de acurácia dos modelos.

E os modelos KNN não demonstram diferenças significativas quando comparados aos SVM mas apresentam quando comparado com os Random-Forest.