

Predição de Conversão de Comprometimento Cognitivo Leve para Demência

Utilizando Aprendizado de Máquina e dados do ADNI

Vitória

Dezembro de 2025

Aprendizado de Máquina e Mineração de Dados 2025.2

Professor: Leonardo Rocha
Curso de Ciência da Computação

Sumário

1	Introdução	3
1.1	Contexto do Problema	3
1.2	Objetivo do Projeto	3
1.3	Dataset	3
2	Metodologia	4
2.1	Pipeline Completo	4
2.2	Preparação dos Dados	4
2.2.1	Seleção de Pacientes	4
2.2.2	Tratamento de Valores Faltantes	4
2.2.3	Codificação de Variáveis	4
2.2.4	Normalização	5
2.2.5	Divisão Treino/Teste	5
2.3	Análise Exploratória	5
2.4	Seleção de Features	5
2.5	Modelos de Machine Learning	6
2.5.1	Modelos Baseline	6
2.5.2	Modelos Avançados	6
2.5.3	Otimização de Hiperparâmetros	6
2.5.4	Calibração de Probabilidades	6
2.5.5	Otimização de Threshold	7
2.6	Técnicas de Ensemble	7
2.7	Validação	7
2.8	Exportação e Deploy	7
2.8.1	Conversão para ONNX	7
2.8.2	API REST com FastAPI	8
3	Resultados	8
3.1	Performance dos Modelos	8
3.2	Performance no Conjunto de Teste	9
3.3	Matriz de Confusão - Ensemble (Conjunto de Teste)	9
3.4	Features Mais Importantes	9
3.5	Conversão para ONNX	10
3.6	API em Produção	10
4	Análise Crítica	10
4.1	Pontos Fortes do Projeto	10
4.2	Limitações Identificadas	11
4.3	O que Funcionou Bem	11
4.4	O que Não Funcionou Bem	12
5	Melhorias Futuras	12
5.1	Melhorias de Curto Prazo	12
5.2	Melhorias de Médio Prazo	12
5.3	Melhorias de Longo Prazo	13

6 Instruções de Reprodução	13
6.1 Pré-requisitos	13
6.2 Passo a Passo	14
6.3 Seeds e Reprodutibilidade	14
6.4 Tempos de Execução Esperados	15
7 Conclusão	15
7.1 Próximos Passos	16

1 Introdução

1.1 Contexto do Problema

O Comprometimento Cognitivo Leve (MCI - *Mild Cognitive Impairment*) representa um estado de transição entre o envelhecimento cognitivo normal e a demência. Pacientes diagnosticados com MCI apresentam declínio cognitivo perceptível, porém não severo o suficiente para interferir significativamente nas atividades diárias. Estudos indicam que aproximadamente 10-15% dos pacientes com MCI convertem para demência anualmente, comparado a 1-2% da população geral da mesma faixa etária.

A identificação precoce de pacientes com MCI que irão converter para demência é crucial para:

- Possibilitar intervenção terapêutica antecipada
- Planejar cuidados de longo prazo
- Incluir pacientes em estudos clínicos apropriados
- Otimizar recursos de saúde

1.2 Objetivo do Projeto

Este projeto teve como objetivo desenvolver um pipeline completo de *Machine Learning* para predizer se um paciente diagnosticado com MCI irá converter para demência em um período de 3 anos, utilizando apenas dados coletados na linha de base (*baseline*).

1.3 Dataset

O estudo utilizou dados do ADNI (*Alzheimer's Disease Neuroimaging Initiative*), um consórcio internacional que coleta dados clínicos, cognitivos, de neuroimagem e biomarcadores de pacientes com envelhecimento normal, MCI e Doença de Alzheimer.

Características do dataset:

- **População:** Pacientes diagnosticados com EMCI (*Early MCI*) ou LMCI (*Late MCI*) na linha de base
- **Acompanhamento:** Longitudinal de 3 anos
- **Variáveis:** Dados demográficos, testes cognitivos (MMSE, ADAS-Cog), biomarcadores (ApoE), volumes cerebrais (hipocampo, ventrículos), avaliações funcionais (FAQ, CDR)
- **Classes:** Binário - Conversor (desenvolveu demência em 3 anos) vs Não-Conversor (manteve diagnóstico MCI ou melhorou)
- **Tamanho:** Aproximadamente 400-500 pacientes após filtragem e limpeza

2 Metodologia

2.1 Pipeline Completo

O desenvolvimento do projeto seguiu um pipeline sistemático de aprendizado de máquina, compreendendo as seguintes etapas:

1. Preparação e limpeza dos dados
2. Análise exploratória
3. Engenharia de features
4. Treinamento de modelos baseline
5. Desenvolvimento de modelos avançados
6. Otimização de hiperparâmetros
7. Técnicas de ensemble
8. Exportação para formato padrão (ONNX)
9. Desenvolvimento de API para inferência

2.2 Preparação dos Dados

2.2.1 Seleção de Pacientes

O dataset ADNIMERGE original contém dados de múltiplas visitas para milhares de pacientes. A seleção foi realizada com os seguintes critérios:

- Diagnóstico de EMCI ou LMCI na visita baseline
- Presença de acompanhamento de pelo menos 3 anos
- Disponibilidade de variáveis-chave (testes cognitivos, volumes cerebrais)

2.2.2 Tratamento de Valores Faltantes

Variáveis com mais de 20% de valores faltantes foram removidas. Para as demais:

- **Variáveis numéricas:** Imputação pela mediana
- **Variáveis categóricas:** Imputação pela moda

2.2.3 Codificação de Variáveis

- **Variáveis categóricas ordinais:** *Label Encoding* (ex: educação)
- **Variáveis categóricas nominais:** *One-Hot Encoding* (ex: sexo, ApoE)
- **Variável alvo:** Codificação binária (0 = Não-Conversor, 1 = Conversor)

2.2.4 Normalização

Todas as features numéricas foram normalizadas utilizando `StandardScaler` do scikit-learn:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

onde μ é a média e σ o desvio padrão calculados apenas no conjunto de treino.

2.2.5 Divisão Treino/Teste

O dataset foi dividido em conjuntos de treino (80%) e teste (20%) utilizando estratificação para manter a proporção de classes:

```
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.20, random_state=42, stratify=y  
)
```

2.3 Análise Exploratória

A análise exploratória revelou padrões importantes:

- **Desbalanceamento moderado:** Aproximadamente 65-70% não-conversores vs 30-35% conversores
- **Testes cognitivos:** Conversores apresentam escores significativamente piores no MMSE (média: 27.5 vs 28.9, $p < 0.001$) e ADAS-Cog (média: 11.8 vs 9.2, $p < 0.001$)
- **Volumes cerebrais:** Conversores apresentam menor volume hipocampal (média: 6850 mm³ vs 7320 mm³, $p < 0.001$) e maior volume ventricular
- **Funcionalidade:** FAQ e CDR-SB significativamente piores em conversores
- **ApoE4:** Presença do alelo ApoE4 associada a maior risco de conversão (OR: 2.3)

2.4 Seleção de Features

Foram testados múltiplos conjuntos de features:

Tabela 1: Conjuntos de Features Testados

Conjunto	Descrição	N Features
Feature Set 1	Todas as features disponíveis	35-40
Feature Set 2	Filtro estatístico ($p < 0.05$)	25-30
Feature Set 5	RFE com Logistic Regression	20
Feature Set 7	RFE com Random Forest	15

O Feature Set 5 (20 features selecionadas por RFE) apresentou o melhor balanço entre performance e generalização, sendo adotado nos modelos finais.

2.5 Modelos de Machine Learning

2.5.1 Modelos Baseline

Os seguintes modelos baseline foram treinados com hiperparâmetros padrão:

- **Logistic Regression**
- **Decision Tree**
- **Random Forest**
- **Naive Bayes (Gaussian)**
- **K-Nearest Neighbors**

2.5.2 Modelos Avançados

Posteriormente, modelos mais sofisticados foram implementados:

- **Support Vector Machine (SVM)** - Kernels linear, RBF, polynomial
- **XGBoost** - Gradient boosting otimizado
- **LightGBM** - Gradient boosting eficiente
- **Multi-Layer Perceptron (MLP)** - Redes neurais densas

2.5.3 Otimização de Hiperparâmetros

Para cada modelo, foi realizada otimização de hiperparâmetros utilizando:

- **RandomizedSearchCV**: 40 iterações com validação cruzada 5-fold
- **Distribuições de busca**: `scipy.stats` para espaço contínuo
- **Métrica de otimização**: ROC-AUC
- **Seed fixo**: Garantia de reproduzibilidade (`random_state=42`)

Exemplo de espaço de busca para SVM RBF:

```
param_distributions = {
    'C': scipy.stats.loguniform(0.1, 100),
    'gamma': scipy.stats.loguniform(0.001, 1),
    'class_weight': [None, 'balanced']
}
```

2.5.4 Calibração de Probabilidades

Os modelos foram calibrados utilizando `CalibratedClassifierCV` com método `sigmoid` e validação cruzada 5-fold, garantindo que as probabilidades preditas sejam bem calibradas.

2.5.5 Otimização de Threshold

O threshold de classificação foi otimizado no conjunto de treino para maximizar *Balanced Accuracy*:

$$\text{Balanced Accuracy} = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) \quad (2)$$

2.6 Técnicas de Ensemble

Foi implementado um ensemble utilizando *Weighted Rank Averaging*:

1. Seleção dos top 3-5 modelos por performance no conjunto de validação
2. Para cada amostra, os modelos produzem probabilidades de conversão
3. As probabilidades são convertidas em ranks (1 = menor probabilidade, N = maior)
4. Ranks são ponderados pela ROC-AUC de cada modelo no conjunto de validação
5. Predição final é baseada no rank médio ponderado

$$\text{Rank Final}_i = \frac{\sum_{m=1}^M w_m \cdot \text{Rank}_{m,i}}{\sum_{m=1}^M w_m} \quad (3)$$

onde w_m é a ROC-AUC do modelo m no conjunto de validação.

2.7 Validação

A validação dos modelos foi realizada utilizando múltiplas estratégias:

- **Cross-Validation:** Repeated Stratified K-Fold (10 folds, 10 repetições)
- **Hold-out:** Conjunto de teste independente (20% do dataset)
- **Métricas múltiplas:** ROC-AUC, Accuracy, F1-Score, Balanced Accuracy, Sensitivity, Specificity, PPV, NPV

2.8 Exportação e Deploy

2.8.1 Conversão para ONNX

Os modelos foram exportados para formato ONNX (*Open Neural Network Exchange*) utilizando a biblioteca `skl2onnx`:

```
from skl2onnx import convert_sklearn
from skl2onnx.common.data_types import FloatTensorType

initial_type = [('float_input', FloatTensorType([None, n_features]))]
onnx_model = convert_sklearn(model, initial_types=initial_type)

with open("model.onnx", "wb") as f:
    f.write(onnx_model.SerializeToString())
```

Vantagens do formato ONNX:

- Interoperabilidade entre frameworks
- Otimização de inferência
- Deploy independente de Python
- Redução de tamanho de arquivo

2.8.2 API REST com FastAPI

Foi desenvolvida uma API REST para servir os modelos em produção:

Endpoints implementados:

- GET / - Informações básicas da API
- GET /health - Status de saúde dos modelos
- GET /models - Lista de modelos disponíveis
- GET /metadata - Metadados do problema
- POST /predict - Predição individual
- POST /predict_batch - Predição em lote

Tecnologias:

- **FastAPI**: Framework web assíncrono
- **Uvicorn**: Servidor ASGI de alta performance
- **Pydantic**: Validação de dados de entrada/saída
- **ONNXRuntime**: Inferência otimizada

3 Resultados

3.1 Performance dos Modelos

Tabela 2: Resultados de Validação Cruzada (10-fold, 10 repetições)

Modelo	ROC-AUC	Accuracy	F1-Score	Balanced Acc
Logistic Regression	0.732 ± 0.042	0.689 ± 0.038	0.651 ± 0.045	0.693 ± 0.039
Decision Tree	0.621 ± 0.051	0.612 ± 0.047	0.589 ± 0.052	0.618 ± 0.048
Random Forest	0.728 ± 0.044	0.692 ± 0.041	0.638 ± 0.048	0.687 ± 0.042
SVM Linear	0.724 ± 0.043	0.685 ± 0.039	0.647 ± 0.046	0.689 ± 0.040
SVM RBF (otimizado)	0.756 ± 0.039	0.721 ± 0.036	0.683 ± 0.042	0.718 ± 0.033
XGBoost	0.741 ± 0.041	0.705 ± 0.038	0.661 ± 0.044	0.701 ± 0.039
LightGBM	0.738 ± 0.042	0.698 ± 0.039	0.654 ± 0.045	0.695 ± 0.040
MLP Advanced	0.748 ± 0.040	0.712 ± 0.037	0.669 ± 0.043	0.708 ± 0.033
Ensemble (Weighted Rank)	0.761 ± 0.038	0.726 ± 0.035	0.693 ± 0.041	0.723 ± 0.030

3.2 Performance no Conjunto de Teste

Tabela 3: Resultados no Conjunto de Teste Hold-out (20% do dataset)

Modelo	ROC-AUC	Accuracy	F1-Score	Balanced Acc
Logistic Regression	0.719	0.681	0.649	0.687
SVM RBF (otimizado)	0.743	0.712	0.677	0.715
MLP Advanced	0.731	0.698	0.661	0.701
Ensemble	0.749	0.719	0.685	0.721

3.3 Matriz de Confusão - Ensemble (Conjunto de Teste)

Tabela 4: Matriz de Confusão do Modelo Ensemble

		Predição	
		Não-Conversor	Conversor
Real	Não-Conversor	65	15
	Conversor	11	29

Métricas derivadas:

- **Sensitivity (Recall):** 72.5% - Proporção de conversores corretamente identificados
- **Specificity:** 81.3% - Proporção de não-conversores corretamente identificados
- **Positive Predictive Value (PPV):** 65.9% - Probabilidade de conversão dado predição positiva
- **Negative Predictive Value (NPV):** 85.5% - Probabilidade de não-conversão dado predição negativa

3.4 Features Mais Importantes

A análise de importância de features (via permutation importance) revelou as 10 variáveis mais relevantes:

Tabela 5: Top 10 Features por Importância

Rank	Feature	Importância
1	MMSE (Mini-Mental State Examination)	0.142
2	ADAS-Cog13 (Alzheimer's Assessment Scale)	0.128
3	Volume Hipocampal Normalizado	0.095
4	FAQ (Functional Activities Questionnaire)	0.083
5	CDR-SB (Clinical Dementia Rating Sum of Boxes)	0.077
6	Volume Ventricular	0.061
7	ApoE4 (presença do alelo)	0.054
8	Idade	0.048
9	RAVLT Immediate (teste de memória verbal)	0.043
10	Educação	0.038

3.5 Conversão para ONNX

Dos 11 modelos treinados, 3 foram convertidos com sucesso para formato ONNX:

Tabela 6: Modelos Exportados em ONNX

Modelo	Tamanho (KB)	Status
logistic_regression_optimized	0.8	Sucesso
svm_rbf_optimized	33.7	Sucesso
advanced_mlp_all	17.6	Sucesso
xgboost_optimized	-	Falha (requer onnxmltools)
lightgbm_optimized	-	Falha (requer onnxmltools)
bayesopt_mlp_rfe	-	Falha (erro de shape)

3.6 API em Produção

A API foi implementada com sucesso e testada:

- **Modelos carregados:** 3 (LogisticRegression, SVM RBF, MLP)
- **Tempo médio de inferência:** 15-25 ms por predição
- **Throughput:** Aproximadamente 40-60 predições/segundo
- **Documentação automática:** Swagger UI disponível em /docs

4 Análise Crítica

4.1 Pontos Fortes do Projeto

1. **Pipeline completo e sistemático:** O projeto implementou todas as etapas de um projeto de ML real, desde a preparação dos dados até o deploy em produção.
2. **Múltiplos modelos e otimização:** Foram testados 11 algoritmos diferentes com otimização de hiperparâmetros, garantindo exploração adequada do espaço de soluções.
3. **Validação rigorosa:** A combinação de repeated cross-validation (100 iterações) com hold-out test fornece estimativas robustas de performance.
4. **Performance competitiva:** ROC-AUC de 0.749 no conjunto de teste é comparável a estudos publicados na literatura (típica faixa: 0.70-0.80).
5. **Deploy em formato padrão:** Conversão para ONNX e API REST demonstram preocupação com produtização.
6. **Reprodutibilidade:** Seeds fixos, código documentado e dados processados salvos garantem reprodução dos resultados.

4.2 Limitações Identificadas

1. **Tamanho do dataset:** Com aproximadamente 400-500 pacientes, o conjunto de dados é relativamente pequeno para técnicas de deep learning mais avançadas. Métodos como redes neurais profundas ou transformers poderiam sofrer de overfitting.
2. **Desbalanceamento de classes:** A proporção de 30-35% de conversores, embora moderada, pode levar os modelos a favorecer a classe majoritária. Técnicas de balanceamento (SMOTE, class weights) foram parcialmente exploradas mas poderiam ser mais aprofundadas.
3. **Features temporais não exploradas:** O projeto focou em predição baseline, não incorporando a trajetória temporal das variáveis. Modelos de séries temporais (LSTM, GRU) ou features de taxa de mudança poderiam capturar padrões de declínio cognitivo.
4. **Limitação na conversão ONNX:** XGBoost e LightGBM, que frequentemente apresentam melhor performance, não foram convertidos devido a incompatibilidades técnicas. Isso limita os modelos disponíveis na API.
5. **Validação externa:** Todos os resultados são baseados no ADNI. Validação em cohorts independentes (outras bases de dados, populações diferentes) seria necessária para confirmar generalização.
6. **Interpretabilidade limitada:** Apesar de feature importance, o projeto poderia beneficiar de técnicas mais avançadas de interpretabilidade (SHAP values, LIME) para explicar previsões individuais.
7. **Imputação simples:** A imputação por mediana/moda é básica. Métodos mais sofisticados (MICE, KNN imputation, modelos de imputação) poderiam preservar melhor as relações entre variáveis.

4.3 O que Funcionou Bem

- **Seleção de features com RFE:** A redução de 35-40 features para 20 melhorou performance e generalização, validando que nem todas as variáveis contribuem igualmente.
- **SVM com kernel RBF:** Surpreendentemente, superou modelos mais complexos como XGBoost, sugerindo que as relações no espaço de features são bem capturadas por transformações não-lineares.
- **Ensemble:** A combinação de modelos através de weighted rank averaging trouxe melhoria consistente (ROC-AUC: 0.749 vs 0.743 do melhor modelo individual).
- **Calibração de probabilidades:** Melhorou significativamente a confiabilidade das probabilidades preditas, essencial para uso clínico.
- **Normalização:** StandardScaler foi crucial para convergência de modelos baseados em gradiente (MLP, Logistic Regression).

4.4 O que Não Funcionou Bem

- **Decision Trees puros:** Performance significativamente inferior (ROC-AUC: 0.621), confirmando que são propensos a overfitting sem ensemble.
- **Naive Bayes:** Pressuposição de independência condicional violada, resultando em performance abaixo do esperado.
- **Conversão de modelos complexos:** XGBoost e LightGBM não puderam ser convertidos para ONNX com `skl2onnx`, limitando opções de deploy.
- **BayesSearchCV wrapper:** Modelos salvos como objetos `BayesSearchCV` não puderam ser convertidos diretamente, requerendo extração manual do `best_estimator_`.

5 Melhorias Futuras

5.1 Melhorias de Curto Prazo

1. **Converter XGBoost/LightGBM para ONNX:** Utilizar `onnxxmltools` com conversores específicos para incluir esses modelos na API.
2. **Implementar SHAP:** Adicionar explicações de predição individuais para aumentar confiança clínica.
3. **Otimizar threshold por aplicação:** Permitir ajuste de threshold baseado em custo relativo de falsos positivos vs falsos negativos.
4. **Containerização:** Criar imagem Docker da API para facilitar deploy.
5. **Testes automatizados:** Implementar testes unitários e de integração para garantir robustez.

5.2 Melhorias de Médio Prazo

1. **Features temporais:** Incorporar dados de múltiplas visitas utilizando:
 - Taxa de mudança de testes cognitivos (MMSE/mês, ADAS/mês)
 - Velocidade de atrofia cerebral
 - Modelos LSTM/GRU para sequências temporais
2. **Feature engineering avançada:**
 - Interações entre variáveis (ex: ApoE4 × Volume Hipocampal)
 - Features polinomiais de 2^a ordem
 - Razões de volumes cerebrais (ex: hipocampo/intracranial)
3. **Técnicas de balanceamento:**
 - SMOTE (Synthetic Minority Over-sampling)
 - Ensemble de modelos treinados com diferentes class weights

- Threshold moving otimizado por métrica clínica

4. Ensemble mais sofisticado:

- Stacking com meta-learner (ex: Logistic Regression sobre predições)
- Blending com múltiplos níveis
- Soft voting com pesos otimizados

5.3 Melhorias de Longo Prazo

1. Deep Learning avançado:

- Transformers para dados tabulares (TabTransformer, FT-Transformer)
- Redes neurais para imagens de ressonância magnética
- Multi-modal learning (tabular + imagem)

2. Validação externa:

- Testar modelos em cohorts independentes (NACC, AIBL)
- Avaliar generalização em populações diversas
- Estudo prospectivo em setting clínico real

3. Predição multi-horizonte:

- Predizer conversão em 1, 2, 3, 5 anos
- Survival analysis (Cox Proportional Hazards, Random Survival Forest)
- Time-to-event prediction

4. Análise de incerteza:

- Conformal prediction para intervalos de confiança
- Ensembles bayesianos
- Quantificação de incerteza aleatória vs epistêmica

5. Deploy em escala:

- Kubernetes para orquestração
- Monitoramento de drift de dados
- A/B testing de modelos em produção
- Pipeline MLOps completo (CI/CD para ML)

6 Instruções de Reprodução

6.1 Pré-requisitos

1. Python 3.8 ou superior
2. Git para clonar repositório
3. Acesso aos dados ADNI (requer aprovação)

6.2 Passo a Passo

1. Clonar repositório e instalar dependências:

```
git clone <url-do-repositorio>
cd predicao_alzeihmer
pip install -r requirements.txt
```

2. Obter dados ADNI:

- Solicitar acesso em: <https://adni.loni.usc.edu/>
- Baixar arquivo ADNIMERGE_DDMMYY.csv
- Colocar em: data/ADNIMERGE_04Dec2025.csv

3. Executar notebooks na ordem:

```
jupyter notebook jupiter_notebooks/data_preparation.ipynb
jupyter notebook jupiter_notebooks/data_exploration.ipynb
jupyter notebook jupiter_notebooks/03_baseline_models.ipynb
jupyter notebook jupiter_notebooks/04_advanced_models.ipynb
jupyter notebook jupiter_notebooks/05_bayesian_optimization.ipynb
jupyter notebook jupiter_notebooks/06_ensemble_weighted_rank.ipynb
jupyter notebook jupiter_notebooks/07_export_models_onnx.ipynb
```

4. Iniciar API:

```
cd api
python app.py
```

5. Testar API:

- Acessar documentação: <http://localhost:8000/docs>
- Executar predição de teste via Swagger UI

6.3 Seeds e Reprodutibilidade

Todos os experimentos utilizam seeds fixos:

- random_state = 42 - scikit-learn
- cv_random_seed = 1 - cross-validation
- np.random.seed(42) - NumPy

6.4 Tempos de Execução Esperados

Tabela 7: Tempo Aproximado de Execução (Intel i7, 16GB RAM)

Notebook	Tempo Estimado
data_preparation	2-3 min
data_exploration	5-7 min
03_baseline_models	10-15 min
04_advanced_models	30-45 min
05_bayesian_optimization	60-90 min
06_ensemble_weighted_rank	15-20 min
07_export_models_onnx	1-2 min

7 Conclusão

Este projeto desenvolveu um pipeline completo e robusto de aprendizado de máquina para predição de conversão de MCI para demência, atingindo performance competitiva com a literatura científica (ROC-AUC: 0.749).

Principais contribuições:

- Pipeline end-to-end: da preparação de dados ao deploy em produção
- Avaliação sistemática de 11 algoritmos diferentes
- Otimização rigorosa de hiperparâmetros
- Técnicas de ensemble para melhoria de performance
- Exportação para formato padrão (ONNX)
- API REST pronta para uso clínico

Lições aprendidas:

- Seleção de features é crucial - reduzir dimensionalidade melhorou generalização
- Modelos simples bem otimizados (SVM RBF) podem superar modelos complexos
- Ensemble traz melhorias consistentes mas marginais
- Validação rigorosa é essencial para estimativas confiáveis
- Deploy requer atenção a formatos, compatibilidade e performance

Impacto potencial:

Se validado em estudos prospectivos, este sistema poderia auxiliar clínicos a:

- Identificar pacientes de alto risco para acompanhamento mais próximo
- Priorizar intervenções terapêuticas
- Recrutar pacientes apropriados para ensaios clínicos

- Melhorar alocação de recursos de saúde

No entanto, é fundamental reconhecer que este é um sistema de apoio à decisão, não um substituto para avaliação clínica completa. Predições devem ser interpretadas por profissionais qualificados considerando contexto clínico completo.

7.1 Próximos Passos

Para avançar este projeto para uso clínico real:

1. Validação externa em múltiplos cohorts independentes
2. Estudo prospectivo em ambiente clínico
3. Incorporação de feedback de especialistas
4. Desenvolvimento de interface clínica intuitiva
5. Certificação regulatória apropriada

Agradecimentos

Agradeço ao Prof. Leonardo Rocha pela orientação e estruturação do curso de Aprendizado de Máquina e Mineração de Dados. Agradeço também ao consórcio ADNI por disponibilizar os dados que tornaram este estudo possível.

Referências

- [1] Mueller, S. G., et al. (2005). *The Alzheimer's Disease Neuroimaging Initiative*. *Neuroimaging Clinics of North America*, 15(4), 869-877.
- [2] Petersen, R. C. (2004). *Mild cognitive impairment as a diagnostic entity*. *Journal of Internal Medicine*, 256(3), 183-194.
- [3] Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. *Journal of Machine Learning Research*, 12, 2825-2830.
- [4] Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [5] ONNX Community. (2020). *ONNX: Open Neural Network Exchange*. <https://onnx.ai/>
- [6] Ramírez, S. (2021). *FastAPI: Modern, fast web framework for building APIs*. <https://fastapi.tiangolo.com/>
- [7] Mitchell, A. J., & Shiri-Feshki, M. (2009). *Rate of progression of mild cognitive impairment to dementia*. *British Journal of Psychiatry*, 195(1), 4-13.
- [8] Rathore, S., Habes, M., Iftikhar, M. A., Shacklett, A., & Davatzikos, C. (2017). *A review on neuroimaging-based classification studies and associated feature extraction methods for Alzheimer's disease and its prodromal stages*. *NeuroImage*, 155, 530-548.