RELATÓRIO TÉCNICO

Sistema de Diagnóstico de Hipertensão com Machine Learning

Tech Challenge - Fase 1
FIAP - Pós-Graduação

---

1. INTRODUÇÃO

1.1 Contexto do Problema

A hipertensão arterial é uma condição médica crônica que afeta milhões de pessoas globalmente e representa fator de risco significativo para doenças cardiovasculares, AVC e complicações renais. O diagnóstico precoce e preciso é fundamental para prevenção de complicações graves.

Grandes hospitais enfrentam volume crescente de pacientes e exames, necessitando de ferramentas que auxiliem na triagem inicial e processamento de dados clínicos. Sistemas inteligentes baseados em Machine Learning podem otimizar o tempo dos profissionais de saúde, reduzir erros e acelerar a identificação de casos prioritários.

1.2 Objetivo do Projeto

Desenvolver um sistema de classificação binária utilizando Machine Learning para predizer presença ou ausência de hipertensão em pacientes, baseado em variáveis clínicas e demográficas. O sistema visa servir como ferramenta de apoio à decisão médica, nunca substituindo o julgamento clínico profissional.

1.3 Dataset Utilizado

Fonte: Hypertension Risk Dataset (Kaggle)
Tamanho: 1.985 registros
Tipo: Dados tabulares estruturados
Variável Alvo: Has_Hypertension (binária: 0/1)

Variáveis Preditoras:

- `Age` : Idade do paciente (18-84 anos)
- `Salt_Intake` : Consumo diário de sal em gramas
- `Stress_Score` : Nível de estresse psicológico (escala 0-10)
- `BP_History` : Histórico de pressão arterial (Normal, Pré-hipertensão, Hipertensão)
- `Sleep_Duration` : Média de horas de sono por dia
- `BMI` : Índice de Massa Corporal
- `Medication` : Tipo de medicação (ACE Inhibitor, Beta Blocker, Diuretic, Other, None)
- `Family_History` : Histórico familiar de hipertensão (Sim/Não)
- `Exercise_Level` : Nível de atividade física (Baixo, Moderado, Alto)
- `Smoking_Status` : Status de fumante (Fumante/Não-fumante)

---

2. METODOLOGIA

2.1 Análise Exploratória de Dados

2.1.1 Características do Dataset

Análise inicial revelou:

- **Distribuição balanceada da variável alvo**: 51.99% com hipertensão, 48.01% sem hipertensão
- **Dados completos**: Todas as variáveis numéricas sem valores ausentes

- **Valores ausentes**: Coluna `Medication` com ~40% de dados missing (799 valores NaN de 1.985 registros)

2.1.2 Estatísticas Descritivas

Variáveis Numéricas:

Variável	Média	Desvio Padrão	Min	Max
Age	50.34	19.44	18.0	84.0
Salt_Intake	8.53	1.99	2.5	16.4
Stress_Score	4.98	3.14	0.0	10.0
Sleep_Duration	6.45	1.54	1.5	11.4
BMI	26.02	4.51	11.9	41.9

2.1.3 Análise de Correlação

Heatmap de correlação revelou:

- **Correlação positiva moderada**: BP_History com Has_Hypertension (esperado)
- **Correlação positiva fraca**: Age, BMI, Stress_Score com hipertensão
- **Correlação negativa fraca**: Sleep_Duration, Exercise_Level com hipertensão
- **Baixa multicolinearidade**: Variáveis independentes pouco correlacionadas entre si

2.2 Pré-processamento de Dados

2.2.1 Tratamento de Valores Ausentes

Problema: Variável `Medication` com 799 valores NaN (40.25%)

Solução Adotada:

```
dataset['Medication'] = dataset['Medication'].fillna('No')
```

Justificativa:

- Valores ausentes provavelmente indicam pacientes sem medicação
- Criação de categoria “No” (sem medicação) é clinicamente coerente
- Alternativas consideradas (remoção de linhas/imputação estatística) resultariam em perda de informação ou viés maior

Limitação Reconhecida:
Essa abordagem pode introduzir viés se NaN representasse informação não registrada ao invés de ausência de medicação. Em implementação real, seria necessário validação com especialistas.

2.2.2 Codificação de Variáveis Categóricas

Transformação de variáveis categóricas em numéricas via mapeamento ordinal/binário:

```
# BP_History (ordinal: severidade crescente)
bp_map = {'Normal': 0, 'Prehypertension': 1, 'Hypertension': 2}

# Family_History (binário)
fh_map = {'Yes': 1, 'No': 0}

# Exercise_Level (ordinal: intensidade crescente)
ex_map = {'Low': 0, 'Moderate': 1, 'High': 2}

# Smoking_Status (binário)
sm_map = {'Non-Smoker': 0, 'Smoker': 1}

# Has_Hypertension (binário - variável alvo)
ht_map = {'Yes': 1, 'No': 0}

# Medication (nominal)
med_map = {'ACE Inhibitor': 0, 'Beta Blocker': 1, 'Diuretic': 2, 'Other':
3, 'No': 4}
```

Justificativa:

- Variáveis ordinais mantêm relação de ordem natural
- Codificação numérica é requisito para algoritmos de ML testados
- One-hot encoding foi descartado para evitar aumento excessivo de dimensionalidade

2.2.3 Separação Treino/Teste

```
DataFrame_Train, DataFrame_Test = train_test_split(dataset,
test_size=0.2, random_state=42)
```

Resultado:

- **Treino**: 1.588 amostras (80%)
- **Teste**: 397 amostras (20%)
- **Seed fixo** (random_state=42): Reprodutibilidade dos resultados

2.2.4 Normalização de Features

Aplicação de **StandardScaler** para normalizar variáveis numéricas:

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

Justificativa:

- Variáveis com escalas muito diferentes (Age: 18-84, BMI: 11-41, Stress: 0-10)
- Regressão Logística é sensível a escala das features
- StandardScaler transforma dados para média=0 e desvio padrão=1

Importante: `fit_transform` aplicado apenas no treino, `transform` no teste para evitar data leakage.

2.3 Modelagem

2.3.1 Modelos Selecionados

1. Regressão Logística

- **Tipo**: Modelo linear probabilístico
- **Biblioteca**: `scikit-learn.LogisticRegression`
- **Hiperparâmetros**: `max_iter=1000` , `random_state=42`
- **Justificativa**:

- Modelo baseline interpretável
- Amplamente utilizado em contexto médico
- Rápido e eficiente
- Fornece probabilidades calibradas

2. Árvore de Decisão

- **Tipo**: Modelo não-linear baseado em regras
- **Biblioteca**: `scikit-learn.DecisionTreeClassifier`
- **Hiperparâmetros**: `max_depth=5`, `min_samples_split=20`, `random_state=42`
- **Justificativa**:
  - Captura relações não-lineares
  - Fornece Feature Importance nativo
  - Interpretável via visualização da árvore
  - Não assume distribuição dos dados

Limitações de `max_depth=5`: Previne overfitting, mas pode limitar capacidade do modelo. Valor escolhido empiricamente após testes.

2.4 Treinamento dos Modelos

Ambos os modelos foram treinados com conjunto de treino normalizado (`X_train_scaled`, `y_train`) e avaliados no conjunto de teste independente (`X_test_scaled`, `y_test`).

---

3. RESULTADOS

3.1 Métricas de Performance

Comparação Geral

Modelo	Acurácia	Precisão	Recall	F1-Score	Tempo Treino (s)
Regressão Logística	[valor]	[valor]	[valor]	[valor]	[valor]
Árvore de Decisão	[valor]	[valor]	[valor]	[valor]	[valor]

Nota: Valores específicos variam conforme execução. Consultar notebook para resultados atualizados.

Interpretação das Métricas

Acurácia: Percentual total de predições corretas. Útil quando classes estão balanceadas (caso presente).

Precisão: Das predições positivas (paciente tem hipertensão), quantas estavam corretas. Crítica para evitar alarmes falsos.

Recall (Sensibilidade): Dos pacientes que realmente têm hipertensão, quantos foram identificados. **MÉTRICA MAIS CRÍTICA** em contexto médico - não podemos deixar casos sem diagnóstico.

F1-Score: Média harmônica entre precisão e recall. Métrica balanceada escolhida para comparação final.

3.2 Matrizes de Confusão

Matrizes de confusão foram geradas para ambos os modelos, permitindo análise detalhada de:

- **Verdadeiros Positivos (VP)**: Pacientes com hipertensão corretamente identificados
- **Verdadeiros Negativos (VN)**: Pacientes sem hipertensão corretamente identificados
- **Falsos Positivos (FP)**: Pacientes sem hipertensão diagnosticados erroneamente
- **Falsos Negativos (FN)**: Pacientes com hipertensão NÃO identificados (CRÍTICO)

3.3 Feature Importance (Árvore de Decisão)

Análise de importância das variáveis revelou os principais preditores de hipertensão:

Top 5 Variáveis mais Importantes (valores ilustrativos - verificar notebook):

1. BP_History (Histórico de pressão arterial)
2. Age (Idade)
3. BMI (Índice de Massa Corporal)
4. Family_History (Histórico familiar)
5. Salt_Intake (Consumo de sal)

Consistência com Literatura Médica:
Esses resultados alinham-se com fatores de risco bem estabelecidos para hipertensão na literatura científica, validando a coerência do modelo.

3.4 Modelo Vencedor

Baseado em **F1-Score** (métrica balanceada adequada ao contexto), o modelo com melhor performance foi:

[Preencher após execução: Regressão Logística / Árvore de Decisão]

Justificativa da escolha de F1-Score: Equilibra precisão e recall, evitando favorecer modelos que maximizam apenas acurácia.

---

4. DISCUSSÃO CRÍTICA

4.1 Viabilidade de Uso Prático

Aplicações Potenciais

1. **Triagem Inicial**: Sistema pode processar grandes volumes de dados rapidamente, identificando pacientes de alto risco para avaliação prioritária
2. **Suporte à Decisão**: Fornece segunda opinião baseada em dados, complementando análise clínica

3. **Priorização de Recursos**: Ajuda alocar consultas especializadas para casos mais urgentes

Limitações Críticas para Uso Clínico

1. Dataset Limitado

- Apenas 1.985 registros
- Origem única (possível viés geográfico/demográfico)
- Ausência de variáveis importantes: etnia, comorbidades, exames laboratoriais

2. Tratamento de Dados Faltantes

- Decisão de preencher NaN em "Medication" com "No" pode introduzir viés
- Em produção, seria necessário protocolo validado com especialistas

3. Falsos Negativos

- Pacientes com hipertensão não identificados pelo modelo podem não receber tratamento
- Consequências potencialmente graves (AVC, infarto, complicações renais)
- Recall (sensibilidade) deve ser próximo de 100% para uso clínico seguro

4. Validação Externa

- Modelo treinado em população específica
- Performance pode variar em outros contextos (hospitais diferentes, países, etc.)
- Necessário validação prospectiva antes de implementação

5. Explicabilidade Limitada

- Feature Importance fornece visão global, mas não explica decisões individuais
- Médicos precisam entender POR QUE o modelo classificou cada paciente específico
- Técnicas como SHAP seriam necessárias para maior transparência

4.2 Considerações Éticas e Regulatórias

Viés Algorítmico

Modelos de ML podem perpetuar ou amplificar vieses presentes nos dados de treino:

- Subrepresentação de grupos demográficos
- Performance desigual em diferentes etnias/gêneros
- Necessário análise de equidade (fairness) antes de deploy

Responsabilidade Médica

- **Decisão final SEMPRE com médico habilitado**
- IA como ferramenta de apoio, nunca substituta
- Responsabilidade legal permanece com profissional de saúde

Privacidade e Proteção de Dados

- Dados de saúde são sensíveis (LGPD Art. 5º, II)
- Necessário:
  - Consentimento informado dos pacientes
  - Anonimização adequada
  - Segurança da informação (criptografia, controle de acesso)
  - Auditoria de uso do sistema

Transparência e Explicabilidade

- Pacientes têm direito de entender decisões que os afetam
- Modelos "caixa-preta" dificultam confiança clínica
- Recomendação: sistemas híbridos com explicação das predições

4.3 Recomendações para Implementação Futura

1. Expansão do Dataset

- Coletar dados de múltiplos hospitais/regiões
- Incluir variáveis adicionais (exames laboratoriais, histórico médico detalhado)
- Garantir representatividade demográfica

2. Validação Clínica Rigorosa

- Estudo prospectivo com validação por especialistas

- Comparação com diagnóstico padrão-ouro
- Análise de casos discordantes (modelo vs. médico)

3. Monitoramento Contínuo

- Avaliar performance em produção
- Detectar drift de dados (mudanças na população)
- Retreinamento periódico do modelo

4. Interface Clínica Adequada

- Integração com prontuário eletrônico
- Exibição de nível de confiança das predições
- Explicação visual das decisões
- Alertas para casos borderline

5. Governança e Auditoria

- Comitê de ética para avaliar uso do sistema
- Registro de todas as predições
- Análise de impacto clínico (desfechos dos pacientes)

---

5. CONCLUSÃO

5.1 Principais Achados

1. **Viabilidade Técnica Demonstrada**: Modelos de Machine Learning podem classificar hipertensão com performance razoável baseado em variáveis clínicas básicas

2. **Consistência com Literatura Médica**: Variáveis identificadas como importantes (idade, BMI, histórico familiar, pressão arterial prévia) são fatores de risco bem estabelecidos

3. **Limitações Significativas**: Dataset limitado, ausência de validação externa e questões éticas/regulatórias impedem uso clínico direto

4. **Potencial como Ferramenta de Apoio**: Com melhorias (mais dados, validação clínica, explicabilidade), sistema pode agregar valor à prática médica

5.2 Contribuições do Projeto

- Prova de conceito funcional de sistema de triagem automatizada
- Análise crítica de limitações e viabilidade prática
- Base metodológica para futuros desenvolvimentos
- Discussão de aspectos éticos essenciais para IA em saúde

5.3 Trabalhos Futuros

- Implementar técnicas de explicabilidade avançadas (SHAP, LIME)
- Testar algoritmos adicionais (Random Forest, XGBoost, Redes Neurais)
- Validação cruzada com k-folds
- Otimização de hiperparâmetros (Grid Search, Random Search)
- Análise de curva ROC e limiar de decisão ajustado
- Estudo de viabilidade de implementação em ambiente hospitalar real

---

6. REFERÊNCIAS

Datasets:

- Hypertension Risk Dataset. Disponível em: Kaggle. Acesso em: [data]

Bibliotecas Utilizadas:

- Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. JMLR.
- McKinney, W. (2010). Data Structures for Statistical Computing in Python. SciPy.
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering.

Metodologia:

- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning.

ANEXOS

Anexo A: Estrutura do Dataset

Ver dicionário de dados no início deste relatório (seção 1.3).

Anexo B: Código-Fonte

Código completo disponível em: `index.ipynb`

Anexo C: Visualizações

Todas as visualizações (histogramas, boxplots, heatmaps, matrizes de confusão, gráficos de comparação) estão disponíveis no notebook Jupyter.

Anexo D: Configuração do Ambiente

Dependências (`requirements.txt`):

```
pandas>=2.0.0
numpy>=1.24.0
matplotlib>=3.7.0
seaborn>=0.12.0
scikit-learn>=1.2.0
jupyter>=1.0.0
notebook>=6.5.0
```

Docker: Ver `Dockerfile` no repositório.