IA: A Nova Esperança na Predição de Diabetes? Uma Análise com Fatores de Risco

Fellipe Gabriel de Oliveira, Jerson Vitor de Paula Gomes, Raul da Cruz Fonseca, Vinicius Ferrer de Queiroz Eloy, Wallace Freitas Oliveira, Cristiane Neri Nobre
Pontifícia Universidade Católica de Minas Gerais
{1313536,1416363,1433894,1314736,1413725}@sga.pucminas.br, nobre@pucminas.br

ABSTRACT

This study explored the use of artificial intelligence (AI) techniques to predict the risk of diabetes based on lifestyle, health, and family history factors. The research utilized data from the 2015 Behavioral Risk Factor Surveillance System (BRFSS) annual survey, managed by the Centers for Disease Control and Prevention (CDC) in the United States.

The data preprocessing involved feature selection, handling of missing values, class balancing, outlier removal, and normalization. Three machine learning models were then applied and compared: Decision Tree, Random Forest, and Multilayer Perceptron (MLP).

The results showed that the MLP model achieved the best performance, with significantly higher precision, recall, and F-score compared to the other models. This superiority was statistically confirmed. The authors conclude that AI can be a valuable tool in predicting diabetes risk, if integrated into medical practice to support early and effective interventions.

KEYWORDS

Diabetes, IA, Machine Learning, KNN Imputer, Isolation Forest, Decision Tree, Random Forest, MLPerceptron, Cross Validation, Machine Learning, Risk Factors, Early Detection, Public Health

1 INTRODUÇÃO

O diabetes mellitus, comumente conhecido como diabetes, é uma doença crônica que afeta a maneira como o corpo regula o açúcar no sangue (glicose) [4]. A glicose é a principal fonte de energia para as células do corpo, e a insulina, um hormônio produzido pelo pâncreas, é responsável por transportá-la da corrente sanguínea para as células [5]. Na diabetes, há um desequilíbrio nesse processo, resultando em níveis elevados de glicose no sangue, condição essa que é conhecida como hiperglicemia. O diabetes é dividida em duas principais categorias e ambas podem levar a sérias complicações de saúde se não forem gerenciadas adequadamente. No diabetes tipo 1, o sistema imunológico ataca por engano as células do pâncreas que produzem insulina, resultando em uma deficiência completa do hormônio. Isso significa que pessoas com diabetes tipo 1 precisam tomar injeções de insulina para toda a vida para controlar seus níveis de glicose no sangue. Já o diabetes tipo 2, que é a forma mais comum da doença, é caracterizado pela resistência à insulina e/ou produção insuficiente de insulina pelo pâncreas. A resistência à insulina significa que as células do corpo não respondem adequadamente à insulina, mesmo quando presente em níveis normais [3]. O diabetes mellitus afeta cerca de 3% da população mundial[6], com prospecto de aumento até 2030, e 50% dos diabéticos desconhecem ter diabete[6], este artigo foi pensado para auxiliar nessa área tendo em vista que o diabete não tratado pode vir a ser letal.

Este artigo apresenta um trabalho de tratamento da base de dados *"2015 Annual Survey Data"* do *"Behavioral Risk Factor Surveillance System (CDC)"*,utilizando diversos métodos para obter resultados que possam ser analisados e relacionados, visando fornecer insights valiosos para o entendimento e o enfrentamento da diabetes, doença essa que afeta milhões de vidas em todo o mundo. O restante do artigo está organizado da seguinte forma, na seção 2 descreveremos os materiais e métodos usados, descrevendo a base de dados usada, e explicando todos os diversos e diferentes métodos usados para tratar os dados. Já na seção 3 discutiremos os resultados obtidos e uma análise detalhada do que foi obtido. Após isso, na seção 4 teremos as considerações finais. Na seção 5 discorreremos sobre a utilização do chat gpt no nosso artigo. E por último na seção 6 estará disponibilizado o link para o código desenvolvido ao longo do projeto.

2 MATERIAL E MÉTODOS

2.1 Base de Dados

A base de dados utilizada neste estudo é a *2015 Annual Survey Data*, compilada pelo *Centers for Disease Control and Prevention* (CDC) por meio de seu programa de vigilância de fatores de risco comportamentais, conhecido como *Behavioral Risk Factor Surveillance System* (BRFSS)[2]. Conduzida em todos os 50 estados dos Estados Unidos da América (EUA), assim como nos distritos de Columbia, Guam e Porto Rico, a pesquisa reuniu um total de 441.456 registros de dados.

Diversos aspectos comportamentais, demográficos e de saúde são contemplados nesta base, fornecendo uma visão abrangente dos fatores de risco que impactam a saúde pública. Dos 330 atributos disponíveis, 20 foram cuidadosamente selecionados para este projeto, levando em consideração a revisão da literatura e sua relevância para os objetivos do estudo, como detalhado na subseção 2.2 Etapas de pré-processamento.

Quanto às características demográficas, os dados revelam uma distribuição significativa: 186.938 instâncias são do sexo masculino e 254.518 do sexo feminino. A faixa etária dos participantes varia de 18 a 65+ anos, com cerca de 72% tendo mais de 45 anos.

Em termos de saúde, destacam-se informações sobre diagnósticos e tratamentos: aproximadamente 160.000 participantes relataram ter sido diagnosticados com colesterol alto, enquanto outros 150.000 fazem uso de medicamentos para pressão arterial.

Avaliando o Índice de Massa Corporal (IMC) e a prática de atividade física, constata-se que 70% dos entrevistados se encontram nas faixas de IMC Normal (1850 <= IMC < 2500) e Sobrepeso (2500 <= IMC < 3000), e que 70% afirmam praticar alguma atividade física regularmente. No entanto, apenas 46% atendem à recomendação da

Organização Mundial da Saúde (OMS) de 150 minutos por semana [10].

Por fim, há também informações sobre o consumo diário de frutas e vegetais: mais de 50% dos entrevistados afirmam consumir 1 ou mais por dia. A organização desses atributos e outros selecionados para o estudo pode ser visualizada na tabela abaixo.

2.2 Etapas de pré-processamento

O processo de pré-processamento de dados é uma etapa fundamental no treinamento de modelos de aprendizado de máquina. Nesse trabalho esse processo foi dividido em 7 etapas principais, conforme ilustrado no fluxograma presente na figura 1, a seguir detalhamos as ações realizadas em cada etapa.

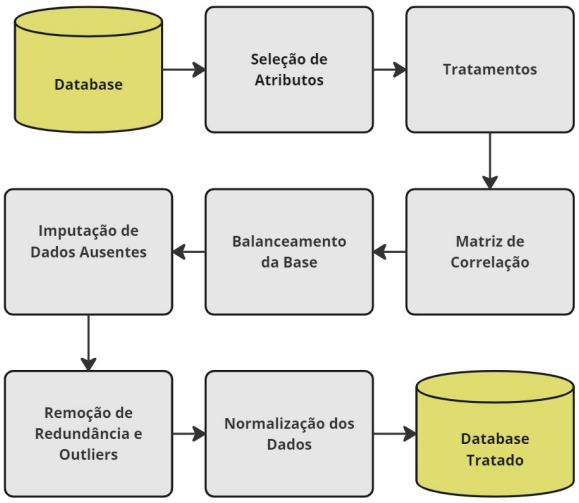

Figure 1: Fluxograma de pré-processamento

2.2.1 ***Seleção Inicial de Atributos.*** A seleção criteriosa de atributos é crucial para o desenvolvimento de modelos de aprendizado de máquina robustos e eficientes, especialmente em áreas complexas como a predição do risco de diabetes. Atributos irrelevantes ou redundantes podem comprometer a precisão e a interpretabilidade do modelo, dificultando a identificação de padrões relevantes e a construção de um modelo preditivo preciso. Para auxiliar na seleção inicial de atributos, recorremos ao *Finnish Diabetes Risk Score* (FINDRISC)[8], um questionário validado e amplamente utilizado para estimar o risco de diabetes tipo 2 nos próximos 10 anos. O FINDRISC se mostrou particularmente útil para nossa pesquisa, pois abrange uma gama completa de fatores de risco relevantes para o desenvolvimento da doença, incluindo aspectos comportamentais, socioeconômicos e genéticos.

Com o auxilio do FINDRISC, conseguimos reduzir o conjunto de atributos inicial de 330 para apenas 20, representando uma diminuição de 94%. Esse redução drástica de atributos otimizou os demais processos de modelagem, resultando em modelos mais eficientes e com menor risco de sobreajuste. A utilização do FINDRISC como auxiliar na escolha dos atributos foi fundamental para confiabilidade de nossa seleção dos fatores de risco a serem abordados na pesquisa, garantindo que o modelo concentrasse nos aspectos mais relevantes para predição do risco de diabetes.

2.2.2 ***Tratamentos.*** A segunda etapa do pré-processamento se concentrou em refinar os dados da base de dados para garantir sua qualidade e confiabilidade. As principais ações realizadas foram:

- **Seleção de instâncias:** No atributo "DIABETE3" selecionamos apenas instâncias com respostas "sim" e "não", garantindo que a análise se concentrasse em indivíduos com diagnóstico confirmado ou negado para diabetes.
- **Correção de valores incoerentes:** Identificamos e corrigimos valores inconsistentes ou que fugiam do significado real dos atributos. Por exemplo, o valor 88 no atributo "PHYSHLTH", que originalmente indicava a ausência de episódios de saúde debilitada nos últimos 30 dias, foi corrigido para o valor correto igual a 0 dias.
- **Tratamento de valores ausentes:** Atributos e instâncias com mais de 20% de seus valores ausentes foram removidos do conjunto de dados. Essa decisão se baseia na premissa de que atributos ou instâncias com alta quantidade de dados ausentes podem conter informações inconsistentes ou irrelevantes para a análise.

2.2.3 ***Matriz de Correlação.*** A análise da matriz de correlação revelou uma forte correlação (90%) entre os atributos "Exerany2" e "_Pai150R2", ambos relacionados à atividade física. Afim de evitar multicolinearidade e assegurar uma otimização para os modelos, a decisão foi manter apenas o atributo "_Pai150R2".

A escolha por manter o atributo "_Pai150R2", se dá pela sua relevância, uma vez que o mesmo avalia a atividade física em relação às diretrizes da OMS [10], um fator crucial para a saúde e prevenção do diabetes. O atributo escolhido considera o nível de atividade semanal enquanto o "Exerany2" se limita à semana anterior à pesquisa.

2.2.4 ***Balanceamento da Base.*** Em face das restrições de hardware da equipe, a priorização do balanceamento da base de dados se sobrepôs à imputação de dados ausentes. A assimetria inicial, com 53.801 instâncias na classe "sim" contra 342.723 na classe "não" do atributo "DIABETE3", exigia uma ação corretiva para otimizar o balanceamento da base.

O algoritmo *Random Under Sampler* [7] da biblioteca **scikit-learn** [15] foi utilizado para balancear as classes. Sua simplicidade e eficiência o tornam ideal para lidar com conjuntos de dados desbalanceados. Seu funcionamento é simples e intuitivo: ele remove aleatoriamente amostras da classe majoritária até que a proporção entre as classes se iguale. Essa ação visa reduzir o viés presente em conjuntos de dados desbalanceados, permitindo que o modelo de *machine learning* aprenda de forma mais justa e precisa.

A inicialização do *Random Under Sampler* permite a passagem do parâmetro **random_state**, no qual o valor 42 foi escolhido. O Resultado final obtido com o algoritmo foi de 53.800 instâncias em cada classe. Na figura 3 é possível observar a otimização obtida com o algoritmo.

Table 1: Descrição dos Atributos

Atributo	Código e Descrição	Valores Possíveis	Tipo de Dado
Alcoolismo	_RFDRHV5: Consumo de álcool	1: Não, 2: Sim, 9: Não classificado	Categórico Nominal
Atividade Física	EXERANY2: Realizou atividade física na última semana	1: Sim, 2: Não, 7: Não sei / Não tenho certeza, 9: Recusou responder	Categórico Ordinal
AVC	CVDSTRK3: Já teve AVC	1: Sim, 2: Não, 7: Não sei / Não tenho certeza, 9: Recusado	Categórico Nominal
Colesterol Alto	TOLDHI2: Diagnóstico de colesterol alto	1: Sim, 2: Não, 7: Não sei / Não tenho certeza, 9: Recusou responder	Categórico Nominal
Consumo de Frutas	_FRTLT1: Consome pelo menos 1 fruta diariamente	1: Uma ou mais, 2: Menos de uma, 9: Não sabe / Não respondeu / Não tem certeza	Categórico Nominal
Consumo de Vegetais	_VEGLT1: Consome pelo menos 1 vegetal diariamente	1: Um ou mais, 2: Menos de um, 9: Não sabe / Não respondeu / Não tem certeza	Categórico Nominal
Diabetes	DIABETE3: Diagnóstico de diabetes	1: Sim, 2: Sim, durante a gravidez, 3: Não, 4: Pré-diabetes ou diabetes limítrofe, 7: Não sei / Não tenho certeza, 9: Recusado	Categórico Nominal
Doença Renal	CHCKIDNY: Apresenta alguma doença renal	1: Sim, 2: Não, 7: Não sei / Não tenho certeza	Categórico Nominal
Doenças Coronárias	CVDCRHD4: Tem angina ou doenças coronárias	1: Sim, 2: Não, 7: Não sei / Não tenho certeza, 9: Recusado	Categórico Nominal
Exame de Rotina	CHECKUP1: Último exame de rotina	1: Há um ano, 2: Há dois anos, 3: Nos últimos cinco anos, 4: Há mais de cinco anos, 8: Nunca realizou, 9: Não sabe / Recusou responder	Categórico Ordinal
Fumante	_SMOKER3: Nível de fumante	1: Fumante, 2: Ocasional, 3: Ex-fumante, 4: Nunca fumou, 9: Recusou responder	Categórico Ordinal
Idade	_AGE_G: Dividida em 6 grupos	1: 18-24, 2: 25-34, 3: 35-44, 4: 45-54, 5: 55-64, 6: 65+	Numérico Discreto
Índice de Massa Corporal	_BMI5CAT: IMC	1: Abaixo do peso, 2: Peso normal, 3: Excesso de peso, 4: Obeso	Categórico Ordinal
Nível Educacional	_EDUCAG: Nível educacional	1: Não concluiu o ensino médio, 2: Graduado no ensino médio, 3: Frequentou faculdade ou escola técnica, 4: Graduado em faculdade ou escola técnica, 9: Não sei / Não tenho certeza / Recusou	Numérico Discreto
Renda Familiar	INCOME2: Renda familiar	1: < $10.000, 2: $10.000-$14.999, 3: $15.000-$19.999, 4: $20.000-$24.999, 5: $25.000-$34.999, 6: $35.000-$49.999, 7: $50.000-$74.999, 8: >= $75.000, 77: Não sei / Não tenho certeza, 99: Recusado	Numérico Discreto
Saúde nos Últimos 30 Dias	PHYSHLTH: Dias com saúde debilitada	1-30: Número de dias, 77: Não sei / Não tenho certeza, 88: Não teve problemas de saúde, 99: Recusou responder	Numérico Discreto
Saúde Geral	GENHLTH: Avaliação da saúde	1: Excelente, 2: Muito bom, 3: Bom, 4: Normal, 5: Ruim, 7: Não sei / Não tenho certeza, 9: Recusado	Categórico Ordinal
Sexo	SEX: Sexo do respondente	1: Masculino, 2: Feminino	Categórico Nominal
Tempo de Atividade Física	_PA150R2: Tempo médio por semana	1: >150 minutos, 2: 1-149 minutos, 3: Não praticou, 9: Não sabe / Não respondeu / Não tem certeza	Categórico Ordinal
Uso de Medicamentos para Pressão	BPMEDS: Realiza uso de medicamentos	1: Sim, 2: Não, 7: Não sei / Não tenho certeza, 9: Recusou responder	Categórico Nominal

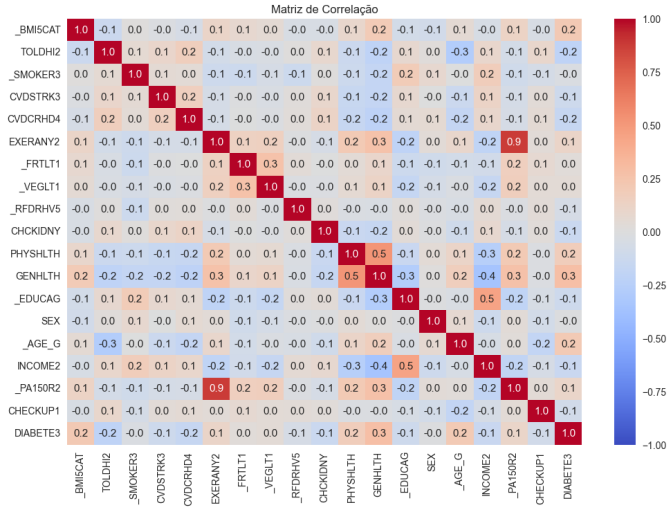

Figure 2: Matriz de correlação da base

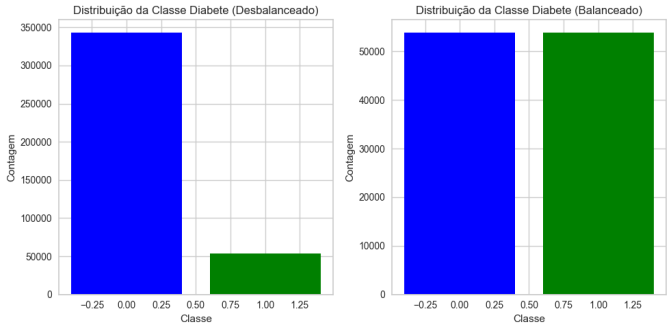

Figure 3: Balanceamento da Base

2.2.5 ***Imputação de Dados Ausentes.*** Após o balanceamento dos dados, identificamos a presença de 2,98% de valores ausentes distribuídos aleatoriamente em diferentes atributos. Para lidar com esse desafio de forma eficaz, optamos por utilizar o algoritmo *KNN Imputer* [11] da biblioteca **Scikit-Learn** [14]. Essa ferramenta robusta se destaca por sua capacidade de imputar valores ausentes preservando a informação original do conjunto de dados, evitando a perda de linhas ou colunas incompletas.

O *KNN Imputer* funciona de maneira inteligente, utilizando o algoritmo *K-Nearest Neighbors* (KNN) para imputar os valores faltantes. Através da busca pelos vizinhos mais próximos no espaço do recurso, ele utiliza a média dos seus valores como estimativa para a lacuna. Essa abordagem garante a preservação da integridade e confiabilidade dos dados, mesmo na presença de valores ausentes.

A versatilidade do *KNN Imputer* o torna ideal para lidar com diversos tipos de dados, incluindo numéricos e categóricos. Ele também é capaz de se adaptar a padrões complexos e relações não lineares entre as variáveis, proporcionando uma imputação precisa e eficiente em diferentes cenários.

Em nosso projeto, configuramos o *KNN Imputer* com o valor de **n_neighbors** igual a 5. Essa escolha visa equilibrar a precisão da imputação com a eficiência computacional, garantindo um bom desempenho geral.

2.2.6 ***Remoção de Redundância e Outliers.*** Após o balanceamento dos dados e a imputação de valores ausentes, a próxima etapa crucial na nossa jornada de pré-processamento envolveu a remoção de outliers e redundâncias. Para lidar com esses valores atípicos que podem distorcer os resultados da análise, aplicamos o algoritmo *Isolation Forest* [9] também presente na biblioteca **Scikit-Learn** [13], e para remoção redundância foi aplicada verificação simples de instâncias repetidas.

O *Isolation Forest* se destaca por sua eficiência e simplicidade na identificação de outliers. Ele funciona de forma análoga a uma floresta de árvores de decisão, isolando os pontos anormais em um processo iterativo. Cada árvore é construída aleatoriamente, dividindo o espaço de dados em subconjuntos até que cada ponto seja isolado em sua própria folha. Em outras palavras, o *Isolation Forest* funciona dividindo os dados em grupos menores até que cada ponto esteja sozinho em um "grupo". Pontos que ficam sozinhos mais rápido são provavelmente anomalias e são removidos.

Em nosso projeto, o *Isolation Forest* foi inicializado usando o parâmetro **contamination** igual a 0.1, permitindo ao algoritmo considerar até 10% dos dados como outliers. No final do processo foram detectados 11.450 outliers.

Para a remoção de redundâncias, realizamos uma verificação simples de instâncias repetidas, que consiste na identificação e eliminação de registros duplicados no conjunto de dados. Essa etapa é essencial para garantir que cada instância no conjunto de dados seja única, evitando que informações repetidas influenciem indevidamente os resultados da análise.

2.2.7 ***Normalização dos Dados.*** A partir da análise dos atributos, observamos que "PHYSHLTH" possui uma grande variedade de valores numéricos distribuídos entre 1 e 30. Essa característica pode prejudicar o desempenho dos modelos preditivos se não for tratada de forma adequada. Para mitigar esse problema, sugerimos a aplicação da normalização utilizando o método MinMaxScaler. Esse método ajusta os dados para um intervalo entre 0 e 1, conforme descrito na equação 1.

$$X_{normalizado} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

Após a normalização, utilizamos um algoritmo adicional chamado OneHotEncoder. Este algoritmo converte atributos categóricos, tanto nominais quanto ordinais, em várias colunas binárias. Essa transformação garante que os modelos de aprendizado de máquina consigam interpretar e aprender com os dados de maneira mais eficiente, ao dividir os atributos em categorias distintas e claras.

2.3 Descrição dos métodos utilizados

O código-fonte deste projeto foi desenvolvido e executado no Google Colab, uma plataforma de computação em nuvem gratuita que fornece acesso a recursos de hardware, como GPUs, facilitando o desenvolvimento e a experimentação de modelos de aprendizado de máquina e todos os algorítimos utilizados pertencem à biblioteca Scikit Learn 1.5 de Maio de 2024.

2.3.1 ***Random Forest.*** Random Forest[1] é um algoritmo de aprendizado de máquina utilizado tanto para tarefas de classificação quanto de regressão. Ele funciona através da construção de um conjunto de árvores de decisão, treinadas em diferentes subconjuntos aleatórios de dados. As previsões finais são feitas por votação majoritária entre as árvores, o que reduz o *overfitting* e melhora a generalização do modelo.

Dentre os hiperparâmetros ajustáveis do *RandomForestClassifier* foram escolhidos afim de otimizar o desempenho do modelo os seguintes:

- **n_estimators:** Define o número de árvores de decisão a serem construídas na floresta. Valores maiores geralmente aumentam a acurácia, mas podem levar a *overfitting*. Valores no Grid: [50, 100, 200]
- **max_depth:** Determina a profundidade máxima das árvores de decisão. Valores maiores permitem que as árvores aprendam relações mais complexas, mas podem aumentar o risco de *overfitting*. Valores no Grid: [10, 20, None]
- **min_samples_split:** Define o número mínimo de amostras necessárias para dividir um nó em uma árvore de decisão. Valores maiores podem levar a árvores mais robustas, mas podem reduzir a precisão. Valores no Grid: [2, 5, 10]

Após testes, os melhores hiperparâmetros encontrados foram:

- **n_estimators:** 200 – Determina que 200 árvores serão construídas na floresta aleatória.
- **max_depth:** 10 - Limita a profundidade das árvores em 10 níveis, evitando *overfitting* e ajudando na generalização.
- **min_samples_split:** 5 - Define que pelo menos 5 amostras são necessárias para realizar uma divisão em um nó da árvore, garantindo uma certa robustez nas árvores.

2.3.2 ***Decision Tree.*** Árvore de Decisão (Decision Tree)[12] é um algoritmo de aprendizado de máquina utilizado tanto para tarefas de classificação quanto de regressão. Ele funcionam através da construção de uma árvore hierárquica que divide os dados em subconjuntos cada vez mais específicos, buscando identificar padrões e regras para prever a classe ou valor de saída para novos dados.

Dentre os hiperparâmetros ajustáveis do *DecisionTreeClassifier* foram escolhidos afim de otimizar o desempenho do modelo os seguintes:

- **max_depth:** Define a profundidade máxima da árvore de decisão. Valores maiores permitem que a árvore capture relações mais complexas nos dados, mas podem aumentar o risco de *overfitting*. Valores no Grid: [10, 20, None]
- **min_samples_split:** Determina o número mínimo de amostras necessárias para dividir um nó na árvore. Valores maiores levam a árvores mais robustas, mas podem reduzir a precisão. Valores no Grid: [2, 5, 10]
- **criterion:** Define o critério utilizado para escolher o melhor split em cada nó da árvore. Valores no Grid:
  - **'gini':** Usa o índice de Gini para medir a impureza de um nó e escolher o split que a reduz mais.
  - **'entropy':** Usa a entropia da informação para medir a impureza e escolher o split que a minimiza.

Após testes, os melhores hiperparâmetros encontrados foram:

- **max_depth:** 10 - Limita a profundidade das árvores em 10 níveis, evitando *overfitting* e ajudando na generalização.
- **min_samples_split:** 5 - Define que pelo menos 5 amostras são necessárias para realizar uma divisão em um nó da árvore, garantindo uma certa robustez na árvores.
- **criterion:** gini - O índice de Gini foi escolhido como a medida de impureza para selecionar o melhor ponto de divisão (split) em cada nó da árvore.

2.3.3 ***ML Perceptron.*** As Redes Neurais Multicamadas (Multilayer Perceptrons - MLPs) são um tipo de rede neural artificial comumente utilizadas para tarefas de classificação. Elas consistem em camadas interconectadas de neurônios artificiais, organizadas em uma estrutura hierárquica:

- **Camada de entrada:** Recebe os dados de entrada (atributos das amostras).
- **Camadas ocultas:** Uma ou mais camadas intermediárias que processam os dados e extraem características complexas.
- **Camada de saída:** Produz a classificação final (previsão da classe para cada amostra).

Dentre os hiperparâmetros ajustáveis do *MLPClassifier* foram escolhidos afim de otimizar o desempenho do modelo os seguintes:

- **hidden_layer_sizes:** Define o número de neurônios em cada camada oculta e a estrutura geral da rede. Valores no Grid: [3, 5]
- **activation:** Define a função de ativação utilizada nos neurônios das camadas ocultas:
  - **'tanh'**: Função tangente hiperbólica, que normaliza os valores de ativação entre -1 e 1.
  - **'relu':** Retificação linear unitária que define a ativação como o valor de entrada se positivo ou zero caso contrário.
- **solver:** Define o algoritmo de otimização utilizado para treinar a rede neural:
  - **'sgd':** Stochastic Gradient Descent (Descida do Gradiente Estocástico), um algoritmo iterativo que atualiza os pesos da rede passo a passo.
  - **'adam':** Adaptive Moment Estimation (Estimativa Adaptativa do Momento), um algoritmo mais recente e geralmente mais eficiente que o SGD.
- **alpha:** Define o parâmetro de regularização L1, que controla a penalização de pesos grandes durante o treinamento. Valores maiores de alpha levam a uma rede mais simples e robusta, mas podem reduzir a precisão. Valores no Grid: [0.0001, 0.05]
- **learning_rate:** Define a taxa de aprendizado, que controla o quanto os pesos da rede são atualizados a cada iteração de treinamento. Valores maiores levam a um treinamento mais rápido, mas podem levar a instabilidade e oscilações na busca pelo melhor conjunto de pesos:
  - **'constant':** A taxa de aprendizado permanece constante durante o treinamento.
  - **'adaptive':** A taxa de aprendizado é adaptada durante o treinamento, geralmente diminuindo com o tempo para evitar sobreajuste.

Após testes, os melhores hiperparâmetros encontrados foram:

- **hidden_layer_sizes:** [3]: A rede utilizará uma única camada oculta com 3 neurônios.
- **activation:** 'tanh': A função de ativação tangente hiperbólica foi escolhida para os neurônios das camadas ocultas.
- **solver:** 'adam': O algoritmo de otimização Adam foi escolhido para treinar a rede.
- **alpha:** 0.05: Este valor define a força da regularização L1.
- **learning_rate:** 'constant': A taxa de aprendizado será mantida constante durante o treinamento.

2.3.4 ***Cross Validation***. Cross Validation[16] é uma técnica para a avaliação de modelos de machine learning que garante capacidade de generalização para novos dados e evita o *overfitting*. Ela consiste em dividir os dados em *folds* e, iterativamente, utilizar cada *fold* para treinamento e validação do modelo. Ela foi utilizada na seleção dos melhores hiperparâmetros para os algoritmos de Random Forest, Decision Tree e MLP. Para cada algoritmo, diferentes configurações de hiperparâmetros foram testadas, e a validação cruzada permitiu identificar a combinação que resultou no melhor desempenho em termos de acurácia para o treinamento dos modelos. Para isso foi utilizado o método de *StratifiedKFold*, uma técnica de validação cruzada estratificada comumente usada em problemas de classificação. O processo de StratifieldKFold é dividido em 3 etapas:

- **Estratificação:** Primeiramente, os dados são divididos em estratos com base na classe alvo (variável de resposta). Isso garante que cada estrato tenha a mesma proporção de amostras de cada classe que o conjunto de dados completo.
- **Separação em Folds:** Em seguida, os estratos são divididos aleatoriamente em K folds (grupos) iguais (ou quase iguais, dependendo do número de amostras por classe).
- **Validação Cruzada:** Na iteração de validação cruzada, cada fold é utilizado como conjunto de validação, enquanto os demais folds combinados formam o conjunto de treinamento. Esse processo se repete K vezes, garantindo que todas as amostras sejam utilizadas para validação e treinamento.

2.4 Métricas de avaliação de qualidade

A qualidade dos modelos desenvolvidos neste estudo foi avaliada através de métricas padrão para classificação binária: precisão, F-score, acurácia e recall, complementadas pelo teste T.

- **Precisão:** Essa métrica revela a confiabilidade do modelo ao identificar elementos positivos. Em outras palavras, indica a proporção de classificações positivas que realmente correspondem à realidade.
- **Recall:** Também conhecido como sensibilidade, o recall mede a abrangência do modelo em encontrar todos os elementos positivos existentes. Ou seja, indica a proporção de elementos positivos reais que foram corretamente identificados pelo modelo.
- **Acurácia:** Abrangendo o desempenho geral do modelo, a acurácia representa a proporção de classificações corretas, tanto para elementos positivos quanto negativos.
- **F-score:** Esta métrica busca um equilíbrio entre precisão e recall, sendo calculada como a média harmônica entre ambas. O F-score é particularmente útil quando se busca um modelo que seja preciso e completo ao mesmo tempo.
- **Teste T:** Completando a análise, o teste T é um método estatístico utilizado para verificar se a diferença entre duas amostras é estatisticamente significante.

3 RESULTADOS E DISCUSSÕES

Os valores de avaliação dos modelos podem ser observados na tabela abaixo

Table 2: Avaliação dos modelos

Modelo	Precisão	Recall	F1-scores
Decision Tree	0.71	0.78	0.75
Random Florest	0.72	0.79	0.76
Multi layer perceptron	0.72	0.79	0.76

Inicialmente, poderíamos afirmar que os resultados são suficientemente próximos para justificar a aplicação do modelo mais simples, uma vez que isso não resultaria em uma perda significativa, enquanto reduziria o custo computacional. No entanto, é possível realizar uma avaliação mais detalhada dos resultados.

Existem algumas avaliações empíricas que podem ser aplicadas. O valor do recall é extremamente importante em modelos de machine learning aplicados na área da saúde, uma vez que é crucial garantir que todos os pacientes que possuem um diagnóstico sejam devidamente indicados a um especialista. No entanto, mesmo com essa consideração, os valores de recall dos modelos são extremamente próximos. Portanto, podemos aplicar outro teste para uma avaliação mais precisa.

O Teste T é utilizado para comparar dois modelos, permitindo avaliar as diferenças entre eles. Os valores resultantes dessa comparação podem ser encontrados na tabela abaixo, ajudando a concluir sobre as diferenças entre os modelos avaliados.

- **Valor T:** Esta métrica representa a performance de um modelo em relação a outro. Valores altos indicam que houve uma diferença significativa entre o desempenho dos modelos avaliados.
- **Valor P:** O valor P indica a relevância estatística da diferença encontrada, sendo qualquer valor abaixo de 0,05 considerado significativo durante a avaliação.

A tabela abaixo apresenta os resultados obtidos a partir dos testes, organizando os nomes dos modelos de forma que o valor T indica a diferença de desempenho do primeiro modelo em relação ao segundo. Conforme pode ser analisado na tabela 3.

Table 3: Resultados Teste T

Modelos	Valor T	Valor P	Significativo?
RF x DT	8.0790	0.0013	Sim
MLP x RF	3.1426	0.0348	Sim
MLP x DT	10.6309	0.0004	Sim

Os valores do teste T indicam que o Random Forest teve um desempenho significativamente melhor que o Decision Tree, e essa diferença de desempenho é estatisticamente relevante. O mesmo ocorre para o Multi-Layer Perceptron e o Random Forest. Portanto,

o MLP obteve os melhores resultados de maneira significativa após a realização dos testes.

Após a avaliação do teste T, torna-se evidente que o Multi-Layer Perceptron funciona significativamente melhor em comparação aos outros dois modelos. Mesmo que inicialmente os valores de desempenho dos modelos pareçam muito semelhantes, o MLP demonstra uma superioridade relevante durante uma análise mais profunda.

4 CONSIDERAÇÕES FINAIS

O objetivo inicial deste projeto era investigar se a inteligência artificial e suas ferramentas poderiam auxiliar na predição de diabetes ao analisar diversos aspectos relacionados aos hábitos, saúde e histórico familiar de um indivíduo. Após várias etapas de processamento, análise e aplicação de diferentes métodos na base de dados *2015 Annual Survey Data*, foi constatada a viabilidade de aplicação de modelos de aprendizado de maquina para esse proposito.

Os atributos utilizados, fundamentados no indicador de risco *Finnish Diabetes Risk Score* [8], permitiram que as predições fossem alinhadas com a literatura médica, aumentando a acurácia das previsões. Embora os resultados iniciais dos modelos fossem bastante próximos, o Multi-Layer Perceptron destacou-se como o mais eficaz após a realização do teste T, validando sua superioridade para nosso objetivo.

A obtenção dos resultados apresentados na seção 3, demonstra que a IA pode sim ser treinada com o objetivo de obter uma predição eficaz do risco de se desenvolver diabetes com base nos dados de comportamento, saúde e histórico familiar, no entanto, é importante destacar que para garantir a segurança dos indivíduos, são necessárias pesquisas mais profundas e abrangentes em diferentes grupos populacionais sobre o tema, para que possamos desenvolver IA's cada vez mais efetiva nessa importante área.

É importante destacar que essas ferramentas devem ser utilizadas como complementos à prática médica, e não de forma isolada, podendo ser implementadas em clínicas ou como ferramentas de diagnóstico precoce. Esperamos que este trabalho inspire novas pesquisas e testes no campo da Inteligência Artificial para a predição de doenças, especialmente aquelas que podem ser prevenidas com mudanças de hábitos. Acreditamos que o avanço neste campo pode proporcionar benefícios significativos à saúde pública, permitindo intervenções mais precoces e eficazes.

5 UTILIZAÇÃO DO GPT

A produção de um artigo científico requer uma análise cuidadosa da escrita, a fim de assegurar o entendimento por parte dos leitores. Neste trabalho, além de explorarmos três diferentes modelos de treinamento de inteligência artificial, buscamos também aproveitar o potencial de dois diferentes modelos de linguagem disponíveis atualmente: o *Generative Pre-trained Transformer*, conhecido popularmente como ChatGPT, e o Gemini. Ambos os modelos foram importantes aliados para correção ortográfica, remoção de vícios linguísticos e outros aspectos fundamentais para a clareza e eficácia na comunicação no artigo.

6 CÓDIGO DESENVOLVIDO

O código desenvolvido nesse projeto, assim como o arquivo da base de dados tratado se encontra disponibilizados para conferência no repositório do GitHub:
https://github.com/JersonVitor/Analise-de-Base-de-dados/tree/main

REFERENCES

[1] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

[2] Centers for Disease Control and Prevention. 2015. 2015 Behavioral Risk Factor Surveillance System Survey Data. https://www.cdc.gov/brfss/annual_data/annual_2015.html

[3] Ministério da Saúde. Disponível em: 08 de Junho de 2024. Diabetes. https://bvsms.saude.gov.br/diabetes/

[4] Ministério da Saúde. Disponível em: 08 de Junho de 2024. Diabetes (diabetes mellitus). https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/diabetes

[5] Núcleo de Produção de Material Educacional NUMEB e Divulgação Científica em Biologia FURG. Disponível em: 08 de Junho de 2024. Regulação da glicemia. https://numeb.furg.br/index.php?option=com_content&view=article&id=42&Itemid=40

[6] Jéssica Muzy e Mônica Rodrigues Campos e Isabel Emmerick e Raulino Sabino da Silva e Joyce Mendes de Andrade Schramm. 2021. Prevalência de diabetes mellitus e suas complicações. https://doi.org/10.1590/0102-311X00076120

[7] Guo Haixiang, Yijing Li, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Bing Gong. 2016. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications* 73 (12 2016). https://doi.org/10.1016/j.eswa.2016.12.035

[8] Jaana Lindstrom and Jaakko Tuomilehto. 2003. The Diabetes Risk Score: A practical tool to predict type 2 diabetes risk. *Diabetes Care* 26, 3 (03 2003), 725–731. https://doi.org/10.2337/diacare.26.3.725 arXiv:https://diabetesjournals.org/care/article-pdf/26/3/725/665299/dc0303000725.pdf

[9] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*. 413–422. https://doi.org/10.1109/ICDM.2008.17

[10] World Health Organization. 2018. *ACTIVE: a technical package for increasinghysical activity.* https://www.who.int/publications/i/item/978924151480

[11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2018. Scikit-learn: Machine Learning in Python. arXiv:1201.0490 [cs.LG]

[12] J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning* 1 (1986), 81–106.

[13] Equipe scikit learn. 2024. *Documentação Isolation Forest.* https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

[14] Equipe scikit learn. 2024. *Documentação KNN Imputer.* https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html

[15] Equipe scikit learn. 2024. *Documentação Random Under Sampler.* https://imbalanced-learn.org/stable/references/generated/imbalanced_learn.under_sampling.RandomUnderSampler.html

[16] Mervyn Stone. 1974. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)* 36, 2 (1974), 111–133.