Classificação de Indicadores de Saúde Relacionados ao Diabetes Usando Naive Bayes

1st Ana Sofia da Silva Barbosa
Centro de informática - UFPE
Recife, Brasil
assb@cin.ufpe.br

2nd Ana Maria Cunha Ribeiro
Centro de informática - UFPE
Recife, Brasil
acmr@cin.ufpe.br

3rd Gabriel Valença Mayerhofer
Centro de informática - UFPE
Recife, Brasil
gvm@cin.ufpe.br

4th Heitor Riquelme Melo de Souza
Centro de informática - UFPE
Recife, Brasil
hrms2@cin.ufpe.br

5th Pedro Henrique Santana de Morais
Centro de informática - UFPE
Recife, Brasil
phsm2@cin.ufpe.br

***Index Terms*—Centro de informática(CIn), Naive Bayes, Dataset, Diabetes, modelos, Machine Learning.**

I. Objetivos

O projeto teve como objetivo principal explorar a aplicação do algoritmo de Naive Bayes na classificação de indicadores de saúde relacionados ao diabetes, utilizando o dataset "CDC Diabetes Health Indicators". Ademais, buscou-se comparar o desempenho do Naive Bayes com outros modelos de classificação, como árvores de decisão, random forest e métodos baseados em redes neurais, visando identificar a melhor abordagem para este conjunto de dados.

II. Justificativa

O diabetes é uma condição que afeta milhões de pessoas em todo o mundo, representando uma preocupação significativa para a saúde pública e sendo essencial a identificação precoce da doença. O dataset Behavioral Risk Factor Surveillance System oferece uma coleção abrangente de dados acerca de práticas de saúde preventiva e comportamentos de risco relacionados a doenças crônicas, lesões e doenças infecciosas preveníveis na população adulta, possibilitando o desenvolvimento de ferramentas personalizadas para prevenção de doenças como o diabetes, além de políticas públicas mais eficazes.

III. Metodologia

Sobre o dataset: Os dados utilizados para a construção do modelo de predição de diabetes fazem parte do conjuntos de dados públicos disponibilizados pela Behavioral Risk Factor Surveillance System (BRFSS). Mais especificamente, serão utilizados os dados de pesquisa de 2013, que contám com mais de 400 mil participantes e mais de 300 atributos. Para o desenvolvimento do modelo, foi utilizado um total de 27 atributos relacionados a saúde, cuidados físicos e psicológicos, alimentação e demográfia. Os atributos são descritos em *Table 1*.

Assim, o desenvolvimento de um modelo de predição tem como objetivo prever o estado do atributo de diabetes baseado no estados dos demais 26 atributos.

A. Análise exploratória

A verificação da congruência e estruturação, além da identificação de padrões e relações nos dados do dataset, foi realizada por meio de uma análise exploratória. Este passo garante a qualidade dos dados e permite a formulação de hipóteses a serem testadas com a técnica de predição utilizada.

Qualidade dos dados: Foram realizados tratamentos para a remoção de entradas do dataset com valores inválidos ou ausentes, além de identificar como cada dado é caracterizado, ou seja, se é binário, contínuo ou categórico. Ademais, para garantir a consistência, foi realizada uma verificação de outliers com o objetivo de assegurar o comportamento esperado do modelo.

Padrões e relações: Um estudo sobre o comportamento dos atributos do dataset foi conduzido utilizando métricas como média, desvio padrão e coeficiente de correlação de Pearson (Pearson's r). Isso permite avaliar quais atributos são mais compatíveis com as características assumidas pelo modelo Naive Bayes.

B. Implementação do algoritmo de classificação

O classificador Naive Bayes foi escolhido por sua eficiência computacional e pela capacidade de lidar com grandes datasets. Este método utiliza do teorema de bayes e assume que há independência condicional entre os atributos, por isso é nomeado ingênuo, isto permite reescrever a relação entre a classe y e os atributos x como:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

Onde o numerador representa a probabilidade conjunta da classe y e dos atributos $x_1 ... x_n$ e o denominador representa

a probabilidade total de observar os atributos $x_1 ..., x_n$, independentemente da classe.

Assumir independência condicional permite o modelo utilizar apenas uma quantidade linear de parâmetros, ao contrário do classificador bayesiano que precisaria de uma quantidade exponencial de parâmetros para capturar a relação entre todos os atributos.

Outra característica importante na implementação do classificador é a escolha da distribuição utilizada para representar a probabilidade dos atributos. Dentre as escolhas, serão avaliadas as seguintes distribuições:

1) Distribuição gaussiana: Modela atributos continuos a partir de uma distribuição gaussiana parametrizada pela média (μ) e variância (σ^2), o classificador utilizando a distribuição gaussiana pode ser descrito pela equação:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

2) Distribuição de bernoulli: Assume que cada atributo é descrito por uma variável discreta booleana (verdadeiro ou falso), ou seja, uma variável binária.

3) Distribuição de bernoulli generalizada (categórica): : Estende a ideia da distribuição de bernoulli, permitindo que cada atributo possua duas ou mais categorias que são assumidas mutuamente exclusivas.

4) Distribuição multinomial: é uma distribuição utilizada quando o atributo envolve contagens de resultados em múltiplas categorias após um número fixo de tentativas independentes.

IV. Treinamento

O treinamento do modelo será realizado com o objetivo de assegurar um aprendizado correto e balanceado e a capacidade de generalização, considerando as particularidades do dataset e o desbalanceamento das classes. Inicialmente, é necessário abordar o desbalanceamento do atributo-alvo, que poderia impactar negativamente a performance do modelo, favorecendo a classe majoritária. Para mitigar esse problema, serão aplicadas e comparadas duas estratégias:

1) Oversampling: É realizado gerando novas amostras sintéticas da classe minoritária a partir de combinações entre as instâncias existentes. Essa abordagem aumenta a representatividade da classe menos frequente, permitindo ao modelo aprender melhor suas características sem repetir os dados originais. Além disso, oversampling preserva o máximo de informação no dataset, pois nenhuma amostra da classe majoritária é descartada.

2) Undersampling: É empregado para reduzir o número de instâncias da classe majoritária, garantindo um equilíbrio entre as classes. Essa técnica reduz o viés do modelo ao evitar que ele se concentre excessivamente na classe dominante.

A vantagem do undersampling é simplificar o treinamento e reduzir o tempo computacional, embora possa sacrificar parte das informações disponíveis. Ao comparar as duas estratégias, será possível determinar o melhor candidato para criar um conjunto de dados equilibrado, mantendo uma boa representatividade das duas classes.

Além disso, a divisão dos dados será realizada utilizando a técnica de K-Fold Cross-Validation. Esse método garante uma avaliação mais robusta, dividindo o conjunto de dados em n partes (folds). Em cada iteração, n-1 folds são utilizados para o treinamento, enquanto o fold restante serve como conjunto de validação. Esse processo é repetido para que cada fold seja usado como validação uma vez, permitindo uma análise abrangente e confiável do desempenho do modelo em diferentes subconjuntos dos dados.

V. Experimento

O treinamento do modelo foi realizado com o objetivo de assegurar um aprendizado correto e balanceado, além de garantir a capacidade de generalização, considerando as particularidades do dataset e o desbalanceamento das classes.

A. Abordagem para o Desbalanceamento

Inicialmente, foi necessário abordar o desbalanceamento do atributo-alvo, que poderia impactar negativamente a performance do modelo ao favorecer a classe majoritária. Para mitigar esse problema, foram aplicadas e comparadas duas estratégias principais: oversampling e undersampling.

- Oversampling: Gera novas amostras sintéticas da classe minoritária a partir de combinações entre as instâncias existentes, aumentando sua representatividade sem repetir os dados originais.
- Undersampling: Reduz o número de instâncias da classe majoritária, garantindo um equilíbrio entre as classes e evitando que o modelo se concentre excessivamente na classe dominante.

Ambas as abordagens foram comparadas para determinar a melhor estratégia, levando em consideração suas vantagens e desvantagens, como a preservação das informações e a redução do tempo computacional.

B. Validação Cruzada

A divisão dos dados foi realizada utilizando a técnica de K-Fold Cross-Validation, garantindo uma avaliação mais precisa. Esse método divide o conjunto de dados em *n* partes (*folds*), onde, em cada iteração, *n-1* folds são utilizados para o treinamento, enquanto o fold restante serve como conjunto de validação. Esse processo é repetido até que cada fold seja utilizado como validação uma vez, permitindo uma análise abrangente e confiável do desempenho do modelo.

C. Tipos de Datasets Utilizados

Para avaliar os efeitos do balanceamento, o conjunto de dados foi inicialmente dividido em 75% para treino e 25% para teste. Em seguida, o conjunto de treino foi segmentado em três variações:

- Dataset de treino original (desbalanceado): Utiliza os dados originais sem nenhuma técnica de balanceamento.
- Dataset com oversampling (SMOTEN): Utiliza a técnica de ”Synthetic Minority Over-sampling Technique for

Nominal Features (SMOTEN)", uma variante do SMOTE projetada para dados binários e categóricos. Diferente do SMOTE tradicional, que funciona interpolando valores numéricos, o SMOTEN gera novas amostras com base na frequência das categorias dos vizinhos mais próximos, preservando a estrutura dos dados categóricos.

- Dataset com undersampling (Cluster Centroids): Utiliza a técnica Cluster Centroids, que aplica algoritmos de clustering (como K-Means) para selecionar representações centrais da classe majoritária, reduzindo o número de instâncias dessa classe sem remover informações importantes.

Cada uma dessas versões foi utilizada separadamente para treinar o modelo, aplicando K-Fold Cross-Validation para garantir uma comparação justa e precisa do desempenho do classificador em diferentes cenários de balanceamento dos dados.

VI. Análise de resultados

Para comparar e analisar os resultados e as métricas de um modelo durante seu treinamento, é comum implementar o processo via código, utilizando bibliotecas como Matplotlib e Seaborn. Essas bibliotecas são utilizadas para gerar gráficos que auxiliam na avaliação do desempenho do modelo. Entre eles, é válido citar a matriz de confusão, que exibe de forma clara a quantidade de acertos e erros do modelo ao classificar as instâncias, dividindo-as nas categorias de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos.

A partir dessa matriz, é possível gerar representações gráficas como o mapa de calor, que visualiza essas categorias de maneira intuitiva, facilitando a interpretação dos resultados. Esses gráficos permitem identificar padrões e áreas onde o modelo pode estar errando, contribuindo para ajustes no processo de treinamento. Isso é crucial para otimizar o modelo e decidir se ele está pronto para ser aplicado em dados reais ou se precisa de mais refinamento. Outros gráficos importantes no processo de treinamento são os de precisão e acurácia, que em suma, ajudam a medir a performance geral do modelo ao longo do seu treinamento.

O gráfico de precisão mostra a capacidade do modelo de não classificar erroneamente uma instância negativa como positiva, enquanto o gráfico de acurácia reflete a proporção de previsões corretas (tanto positivas quanto negativas) em relação ao total de previsões feitas. Ambos são indicadores importantes, mas não devem ser analisados isoladamente, pois, em alguns casos, métricas como a precisão podem ser mais relevantes do que a acurácia, especialmente quando os dados estão desbalanceados.

Além disso, gráficos de recall e F1-score são úteis para fornecer uma visão mais detalhada do comportamento do modelo. O recall mede a capacidade do modelo em identificar todas as instâncias positivas, enquanto o F1-score busca balancear precisão e recall, sendo uma boa métrica quando há necessidade de otimizar tanto a taxa de acertos quanto a de erros. Esses gráficos, quando usados em conjunto, ajudam a diagnosticar os pontos fortes e fracos do modelo, fornecendo insights valiosos para ajustar os parâmetros para caso de retreinamento ou até mesmo a escolha do algoritmo de classificação. Com isso, torna-se possível garantir que o modelo seja robusto e eficaz para ser implementado em situações reais de aplicação.

VII. Conclusões

ComplementNB (Desbalanceado): Este modelo obteve uma precisão de 0.7757 e um recall de 0.7358, com um F1-score de 0.7525. Embora a precisão seja relativamente alta, o recall para a classe minoritária (1.0) é de apenas 0.43, o que indica que o modelo ainda tem dificuldades para identificar corretamente os exemplos dessa classe. A acurácia foi de 74%, o que reflete um desempenho razoável, mas a performance desequilibrada entre as classes sugere que melhorias podem ser feitas. Também é observado que devido ao desbalanceamento da base de dados, o modelo tendeu a classificar mais como 0 independente da classe verdadeira. Pode-se constatar pelo "F1-score" e pelo "Precision".

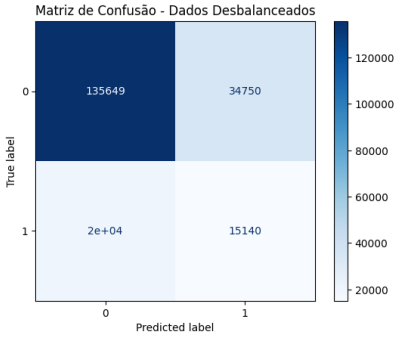

Fig. 1. Matriz de Confusão com Dataset desbalanceado

CategoricalNB (Undersampling): Com a técnica de undersampling, a precisão foi mais alta (0.8230), mas o recall caiu significativamente para 0.5470, resultando em um F1-score de 0.5968. A acurácia foi de 55%, indicando que o modelo perdeu bastante informação da classe majoritária ao reduzir seu tamanho para balancear o dataset. Embora o recall da classe majoritária (0.0) seja razoável, o recall para a classe minoritária aumentou, mas a acurácia geral sofreu devido ao subamostragem.

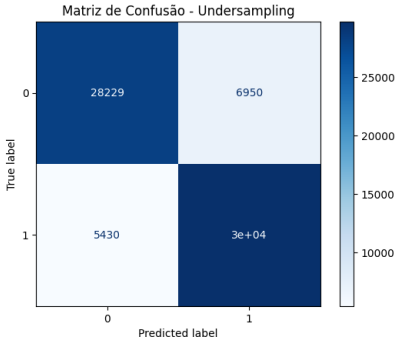

Fig. 2. Matriz de Confusão com o Dataset tratado com 'Undersampling'

CategoricalNB (Oversampling): Ao aplicar oversampling, o modelo obteve um recall de 0.7287, precisão de 0.8165 e F1-score de 0.7570. A acurácia foi de 73%, sendo o melhor desempenho em relação ao undersampling, com uma boa capacidade de identificar as instâncias da classe minoritária sem perder muito da classe majoritária. O modelo teve um bom equilíbrio entre precisão e recall, superando o modelo undresampling em termos de desempenho para ambas as classes.

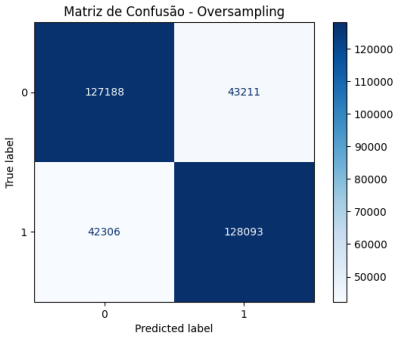

Fig. 3. Matriz de Confusão com o Dataset tratado com 'Oversampling'

REFERENCES

[1] Centers for Disease Control and Prevention (CDC). Behavioral Risk Factor Surveillance System

[2] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

[3] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[4] Vujovic, Zeljko. (2021). Classification Model Evaluation Metrics. International Journal of Advanced Computer Science and Applications. Volume 12. 599-606.

TABLE I
DESCRIÇÃO DOS PARÂMETROS DA BASE DE DADOS

Atributo	Descrição
GenHealth	Autoavaliação do estado geral de saúde de 1 a 5 (Excelente, muito bem, bem, razoável, mal).
phyHealthDays	Número de dias no último mês em que a saúde física esteve prejudicada (0 - 30).
menHealthDays	Número de dias no último mês em que a saúde mental esteve prejudicada (0 - 30).
hasHealthPlan	Indicador binário de posse de plano de saúde (0, 1).
hasPersonalDoctor	Indicador binário de acesso a médico pessoal (0, 1).
diffWalking	Indicador binário de dificuldade para caminhar ou subir escadas (0, 1).
costLimitation	Indicador binário de limitação no acesso a cuidados médicos por custo nos últimos 12 meses (0, 1).
lastCheckup	Tempo desde a última consulta médica (1: menos de 1 ano, 2: entre 1 e 2 anos, 3: entre 2 e 5 anos, 4: mais de 5 anos).
hypertensionRisk	Indicador binário de hipertensão relatado por um profissional de saúde (0, 1).
cholesterolRisk	Indicador binário de colesterol alto relatado por um profissional de saúde (0, 1).
hadHeartAttack	Indicador binário de histórico de infarto (0, 1).
hadCHD	Indicador binário de histórico de doença coronariana (0, 1).
hadStroke	Indicador binário de histórico de AVC (0, 1).
hadDepreDisorder	Indicador binário de diagnóstico prévio de transtorno depressivo (0, 1).
hadKidneyDisease	Indicador binário de histórico de doença renal (0, 1).
gender	Sexo do participante (1: masculino, 2: feminino).
ageGroup	Faixa etária categorizada em intervalos de anos.
educationLevel	Nível educacional categorizado (1: ensino médio incompleto, 2: ensino médio completo, 3: ensino superior ou técnico incompleto, 4: ensino superior ou técnico completo).
employmentStatus	Situação empregatícia (1: Empregado assalariado, 2: autônomo, 3: Desempregado há 1 ano ou mais, 4: Desempregado há menos de 1 ano, 5: Responsável pelo lar, 6: Estudante, 7: Aposentado, 8: Incapaz de trabalhar).
incomeGroup	Faixa de renda anual categorizada (1: Menos de $15.000, 2: De $15.000 a menos de $25.000, 3: De $25.000 a menos de $35.000, 4: De $35.000 a menos de $50.000, 5: $50.000 ou mais).
bmiCategory	Categoria de IMC (1: Abaixo do peso, 2: Peso normal, 3: Sobrepeso, 4: Obesidade).
smokingStatus	Histórico de tabagismo (1: fuma diariamente, 2: fuma algumas vezes, 3: já fumou, 4: nunca fumou).
drinksPerDay	Número médio de doses alcoólicas consumidas por dia.
fruitConsumption	Indicador binário de consumo de fruta diário (1: uma ou mais frutas por dia, 2: menos de uma fruta por dia).
vegetableConsumption	Indicador binário de consumo de vegetais diariamente (1: um ou mais vegetal por dia, 2: menos de um vegetal por dia).
phyActivity	Indicador binário de prática de exercício físico no último mês (0, 1).
diabetes	Indicador binário de diabetes (0: não tem diabetes, 1: tem diabetes ou é pré-diabético).