UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO LATU SENSU EM VISÃO COMPUTACIONAL

OTÁVIO KAMEL
WESLEY SOUZA

ESTIMATIVA DOS NÍVEIS DE OBESIDADE COM BASE EM HÁBITOS ALIMENTARES E CONDIÇÃO FÍSICA

Recife-PE
2022

OTÁVIO KAMEL
WESLEY SOUZA

ESTIMATIVA DOS NÍVEIS DE OBESIDADE COM BASE EM HÁBITOS ALIMENTARES E CONDIÇÃO FÍSICA

Projeto de pesquisa apresentado como requisito à obtenção de aprovação na disciplina de Aprendizagem de Máquinas do Curso de Pós-Graduação Latu Sensu em Visão Computacional da Universidade Federal de Pernambuco.

Prof. Dr. George Darmiton da Cunha Cavalcanti
.

Recife-PE
2022

Sumário

1 Introdução

A obesidade é uma doença definida pela Organização Mundial de Saúde como o acúmulo anormal ou excessivo de gordura que apresenta risco para saúde, desde de 1975 quase triplicou sua ocorrência mundialmente, em 2016 totalizou mais de 650 milhões de pessoas acometidas pela enfermidade, e em 2020 foram constatadas 39 milhões de crianças abaixo de 5 anos com sobrepeso ou obesas (1).

A obesidade contribui para uma redução da expectativa de vida em decorrência do aumento da mortalidade por doenças não transmissíveis, incluindo doenças cardiovasculares ateroscleróticas, diabetes tipo 2 e determinados tipos de câncer. Além das consequências da obesidade no nível individual, a pandemia da obesidade pode causar um grande impacto para a saúde da sociedade, somente nas últimas cinco décadas ela quase triplicou em adultos e aumentou ainda mais em crianças e adolescentes (2).

O presente trabalho tem como objetivo apresentar um modelo de predição capaz de prever os níveis de Obesidade dos indivíduos por meio características extráidas de um dataset que contém dados de condições alimentares, fisícas e comportamentais.

2 Algoritmos de Aprendizagem de Máquina

Os algoritmos de aprendizagem de máquina aplicados nesse trabalho são do tipo supervisionados, visto que a as variáveis alvos (rótulos) estão contidas na base de dados, e o objetivo da classificação é estabelecer um modelo capaz de realizar a predição dos níveis de obesidade dos indivíduos a partir de seus hábitos alimentares e condição física.

O primeiro algoritmo é o k-vizinhos mais próximos, denominado do inglês K-NN (*K Nearest Neighbor*), consiste em utilizar as características para determinar a localização da predição e calcula sua distância para as demais k-predições para conseguir prever a classe pertencente, Nessa estratégia, foi importante ajustar dois parâmetros do algoritmo para obter resultados mais fidedignos com a realidade, o primeiro é o número de k-vizinhos e o segundo é o tipo de métrica usada para calcular a distância, para esse projeto foi adotada a distância de manhattan.

Em seguida a Árvore de Decisão será avaliada para classificação das classes, essa abordagem basicamente estrutura as características em uma estrutura de dados de árvore, considerando os nós mais próximos a raiz os mais importantes seguindo para os nós mais abaixo por meio de análises condicionais para estrutura-lá até as folhas. Como se trata de um algoritmo recursivo e que apresenta bons resultados para base de dados com baixo grau de entropia, se adequa bem ao contexto apresentado. A performance do modelo será discutida com mais detalhes na seção 3.4.

O algoritmo Perceptron Multicamadas, do inglês MLP (*Multi Layer Perceptron*) também será avaliado, esse algoritmo faz uso de camadas de redes neurais, onde cada neurônio faz ajustes ao longo da execução em um sistema de "punição e bonificação"para melhorar o modelo, para

construir esse modelo no projeto foram observadas a quantidade de camadas além do número de iterações máximas realizadas por esse algoritmo.

O modelo construído a partir do algoritmo de Naive Bayes utiliza como base o próprio teorema de Bayes utilizando cálculos probabilísticos para realizar inferências, em níveis de complexidade apresenta-se como um dos mais simples para aprendizagem de máquinas. Por consequência, também apresenta pouca flexibilidade para se adaptar em contextos distintos de informações, já que apresenta dificuldades para correlacionar diferentes características.

O método SVM (*Support Vector Machine*) assemelha-se à construção do K-NN, ao empregar a distância entre os pontos dos dados para realizar a predição. Porém, o seu princípio é diferente para encontrar as soluções, pois ele estabelece um hiper-plano com o intuito de categorizar as classes estabelecendo um limiar para margem de erros que possa ocorrer. A desvantagem desse algoritmo é a quando o conjunto de dados apresenta classes sobrepostas, ou quando a ela é muito grande, isso pode interferir diretamente no tempo de execução.

3 Experimentos

Esta seção irá apresentar em três tópicos a forma como os experimentos foram construídos e conduzidos. A primeira parte explica a origem dos dados e a descrimina os atributos nele contido; a segunda parte mostra as métricas e metodologias adotadas para cada algoritmo usado para realizar as predições; e a ultima etapa discute os resultados obtidos.

3.1 Banco de dados

Para realização desse trabalho foram utilizados dados oriundos do trabalho "**Conjunto de dados para estimativa dos níveis de obesidade com base em hábitos alimentares e condição física em indivíduos da Colômbia, Peru e México**". Este trabalho apresenta dados para a estimativa dos níveis de obesidade em indivíduos dos países do México, Peru e Colômbia com idade entre 14 e 61 anos, com base em seus hábitos alimentares e condição física. O conjunto de dados contém 17 atributos e 2111 registros, que são rotulados com a variável de classe *NObeyesdad*, que representa o nível de obesidade para cada indivíduo. A classificação da classe alvo é feita utilizando os rótulos de Peso Insuficiente, Peso Normal, Sobrepeso Nível I, Sobrepeso Nível II, Obesidade Tipo I , Obesidade Tipo II e Obesidade Tipo III. Os dados foram coletados diretamente dos usuários por meio de uma plataforma *web* (3).

Os atributos relacionados são:

- Consumo frequente de alimentos hipercalóricos (FAVC - *Frequent consumption of high caloric food*)
- Frequência de consumo de hortaliças (FCVC - *Frequency of consumption of vegetables*)
- Número de refeições principais (NCP - *Number of main meals*),

- Consumo de alimentos entre as refeições (CAEC - *Consumption of food between meals*)
- Consumo diário de água (CH20 - *Consumption of water daily*)
- Consumo de álcool (CALC - *Consumption of alcohol*)
- Monitoramento do consumo de calorias (SCC - *Calories consumption monitoring*)
- Frequência de atividade física (FAF - *Physical activity frequency*)
- Tempo de uso de dispositivos tecnológicos (TUE - *Time using technology devices*)
- Transporte utilizado (MTRANS - *Transportation used*)
- Fumante (*SMOKE*)
- Gênero (*Gender*)
- Idade (*Age*)
- Altura (*Weight*)
- Peso (*Height*)

Após os dados serem rotulados, foi notado que as classes estavam desbalanceadas. Um conjunto de dados é considerado desbalanceado se as categorias de classificação da variável alvo possui um número de observações muito discrepantes entre si, conforme pode ser visualizado na Figura 1, fator que pode interferir diretamente no aprendizado de máquina. Devido a esse fato, foram gerados dados sintéticos utilizando a ferramenta *Weka* e o filtro *SMOTE* seguindo a proposta de (4).

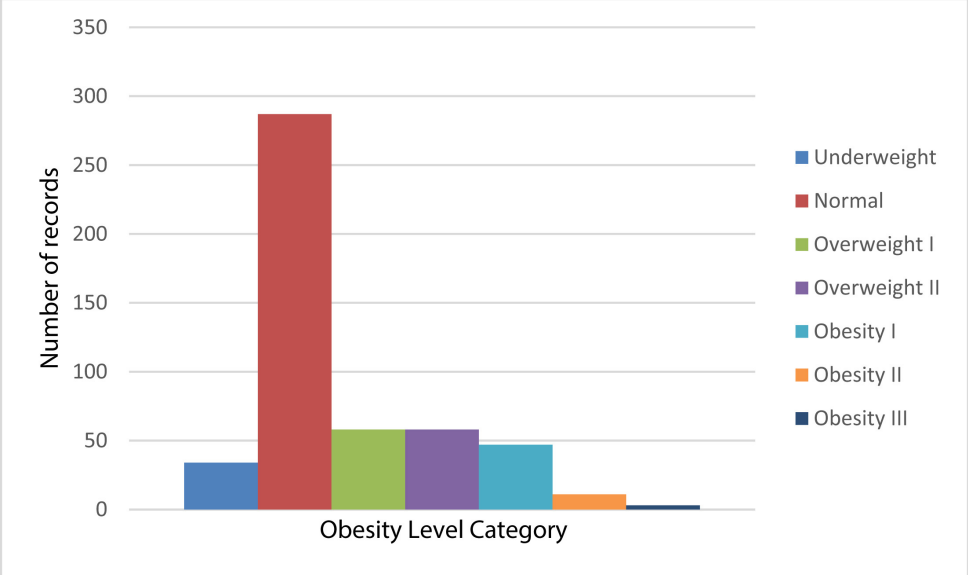

Figura 1. *Distribuição não balanceada de dados em relação à categoria níveis de obesidade*

Após a aplicação do filtro em cada categoria, obteve-se ao final 2111 registros onde 77% desses dados foram gerados sinteticamente, resultando em conjunto de dados balanceados como pode ser observado na Figura 2.

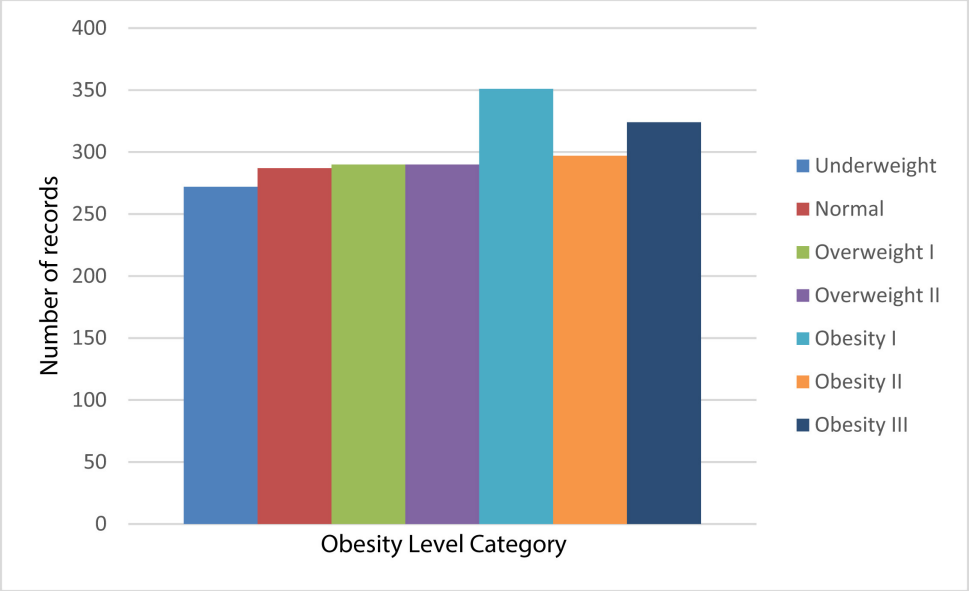

Figura 2. *Distribuição balanceada dos dados referentes à categoria de níveis de obesidade*

3.2 Análise e Visualização dos Dados

A primeira etapa de trabalho iniciou com o uso da linguagem Pyhton na versão 3.8, para a coleta, limpeza e visualização dos dados, além da aplicação dos modelos de aprendizagem de máquina já citados no presente trabalho na seção 2. A partir da análise de estatística descritiva geral sobre o conjunto de dados, que pode ser visto na Figura 3, foi possível identificar características da amostra populacional, como por exemplo:

	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE	
count	2111	2111	2111	2111	2111	2111	2111	2111	2111
mean	24.31	1.7	86.59	2.42	2.69	2.01	1.01	0.66	
std	6.35	0.09	26.19	0.53	0.78	0.61	0.85	0.61	
min	14	1.45	39	1	1	1	0	0	
25%	19.95	1.63	65.47	2	2.66	1.58	0.12	0	
50%	22.78	1.7	83	2.39	3	2	1	0.63	
75%	26	1.77	107.43	3	3	2.48	1.67	1	
max	61	1.98	173	3	4	3	3	2	

Figura 3. *Medidas Descritivas dos Dados*

- A idade dos indivíduos está em um intervalo entre 14 e 61 anos
- Apenas 1/4 dos indivíduos encontra-se numa faixa de idade abaixo de 20 anos

- Mais de 25% apresentou peso corporal acima de 100kg

Posteriormente, os dados de cada característica foram dispersos em histogramas, para poder obter uma visualização dos dados de forma gráfica para compreender o comportamento deles individualmente. Os gráficos gerados estão dispostos na Figura 4.

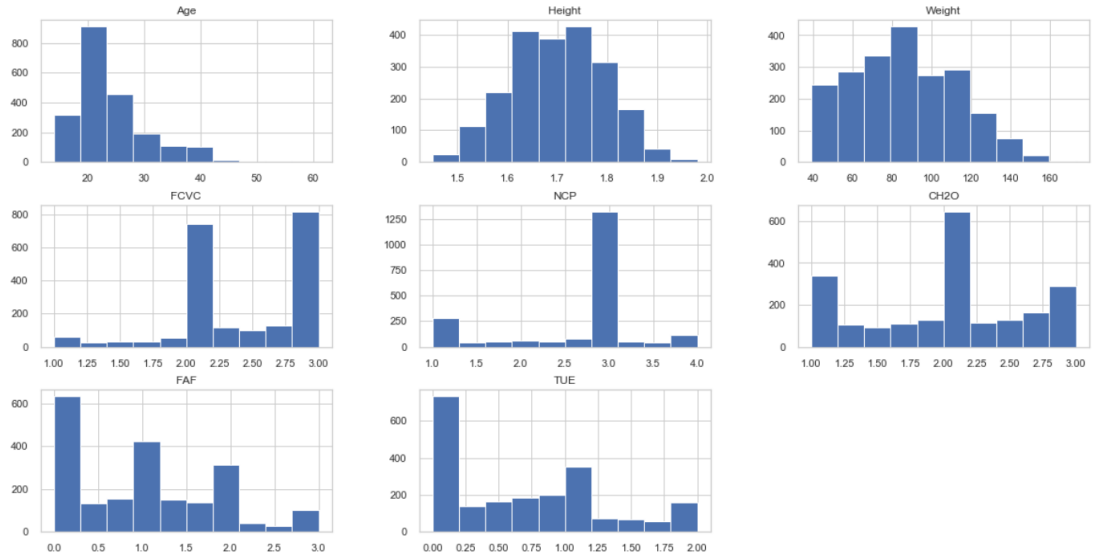

Figura 4. *Histogramas das características da base de dados*

Com a visualização desses histogramas, é possível denotar que a idade possui uma concentração de dados entre 20 e 30 anos, o que permite prever que os dados possuem um viés relacionado a faixa etária. Além disso, o Consumo de refeições diárias e o Consumo diário de água também têm valores concentrados respectivamente em 3 refeições e 2 litros: isso pode ser explicado mediante a forma de obtenção dos dados por meio de pesquisa por formulário na *web* e não utilizando valores aferidos experimentalmente.

A matriz de correlação das características conforme demonstrado na Figura 5 apresenta uma forte correlação entre os dados de Altura e Peso, e correlações mais fracas entre Altura com outras duas variáveis, a Frequência de atividade física e Número de refeições principais.

Após essas análises e visualizações, foi verificado o intervalo de distribuição de cada atributo do dataset através de um diagrama de caixa (boxplot), para decidir a aplicação do tipo de escalonamento em cada um deles, quando necessário. Nas variáveis em que se observou muitos outliers, foi utilizado o método RobustScaler da biblioteca scikit-learn, enquanto nos restantes usou-se o MinMaxScaler. Esse processo irá permitir que os dados passados como entrada para o modelo de machine learning não pertubem o aprendizado dos algoritmos por não estarem em uma escala de valor próxima. Além disso, as variáveis categóricas foram convertidas para numéricas através de encoding, pois a maioria dos algoritmos de aprendizado de máquina não conseguem trabalhar com dados não numéricos. Os métodos de conversão utilizados foram o label encoding, ordinal encoding e o frequency encoding.

Por último as caraterísticas mais e menos relevantes foram identificadas, a fim de tornar os modelos mais assertivos possível. Para realizar essa tarefa, foi utilizada uma biblioteca do

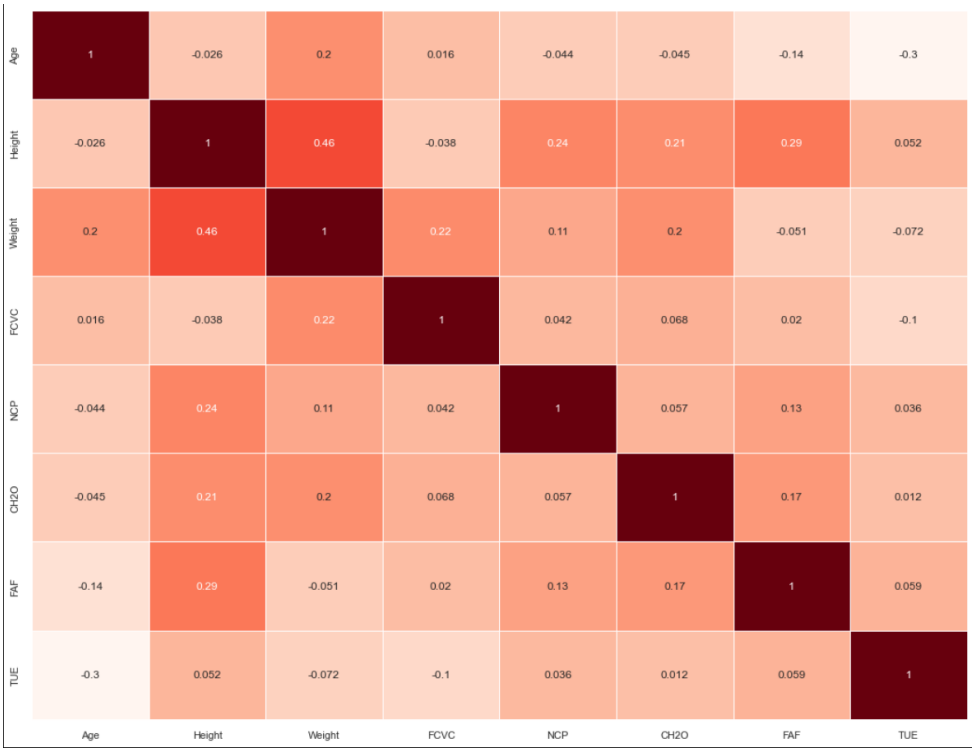

Figura 5. *Matriz de correlação das características do modelo*

Python chamada Boruta, que tenta capturar todos recursos importantes e interessantes da base de dados relacionados aos resultados. No Boruta, as características são avaliadas com uma versão aleatória delas mesmas, baseado na mesma ideia que forma a base do classificador *Random Forest*. Essa aleatoriedade extra dá uma visão mais clara de quais atributos são realmente importantes (5). Ao final dessa análise, as colunas referentes as informações de Consumo frequente de alimentos hipercalóricos, Transporte utilizado, Monitoramento do consumo de calorias e Fumante se mostraram menos relevantes para construção dos modelos.

3.3 Métricas utilizadas

As medidas de desempenho são métricas que ajudam a compreender o nível de performance dos resultados obtidos com os algoritmos de aprendizagem apresentados na seção 2; são análises aplicadas que revelam diferentes aspectos dos modelos. A métrica mais intuitiva é a acurácia que revela a taxa de acerto das predições e é dada pela equação 1.

$$\text{acurácia} = \frac{\text{Numero de previsões corretas}}{\text{Numero total de previsões}} \tag{1}$$

Entretanto essa métrica inicial embora muito interessante, não é capaz de abranger a completude de desempenho dos modelos. Dessa forma, a de matriz de confusão é um conceito importante para o compreendimento das demais métricas, e se baseia em uma matriz *NxN* onde cada linha representa uma classe real e cada coluna representa uma classe prevista. Essa matriz apresenta valores de verdadeiros-positivos *VP*, falsos-positivos *FP*, verdadeiros-negativos *VN* e

falsos-negativos *FN*. Esses valores são usado para calcular a precisão e revogação demonstrados respectivamente nas equações 2 e 3.

$$\text{precisão} = \frac{\text{VP}}{\text{VP + FP}} \tag{2}$$

$$\text{revocação} = \frac{\text{VP}}{\text{VP + FN}} \tag{3}$$

Esses valores são fundamentais por complementarem os resultados que somente a acurácia não é capaz de evidenciar, e em conjunto são capazes de estabelecer uma métrica mais abrangente, uma média harmônica entre a precisão e revocação denominada *f1-score* que é exibido na equação 4

$$\text{f1-score} = 2 \cdot \frac{\text{precisão} \cdot \text{revocação}}{\text{precisão} + \text{revocação}} \tag{4}$$

Ainda fazendo uso do conceito da matriz de confusão, outra métrica relevante que pode ser estabelecida é a curva ROC (*Receiver Operating Characteristic Curve*) e o valor da área sob a curva (AUC - *Area Under the Curve*). A curva ROC traça a taxa de verdadeiros positivos pela taxa de falsos positivos em diferentes limiares de classificação, enquanto que a AUC resume a curva ROC em um único valor, calculando a área sob a curva.

3.4 Resultados e discussões

Inicialmente, o conjunto de dados foi dividido entre conjunto de treino, validação e de teste: 10% do total para teste (212 registros); e os 90% restantes foram separados em treino e validação, sendo 80% das observações para treino (1519 registros) e 20% para validação (380 registros). Após a realização do escalonamento e encoding, foi criada uma função para dividir o conjunto de treino em 10 grupos estratificados (stratified k-folds, que contêm o número de classes praticamente igual entre os diferentes grupos), para em seguida treinar e validar os dados para cada algoritmo, e por fim, imprimir na tela os valores referentes às métricas mencionadas na subseção 3.3. O próximo passo foi utilizar o grid search para ajustar os hiperparâmetros dos modelos de machine learning, a fim de otimizar os seus resultados. Os hiperparâmetros percorridos e os melhores resultados para os algortimos estão dispostor a seguir, com os hiperparâmetros escolhidos em negrito:

K-NN:

- número de vizinhos (de 2, 3, **4**, 5, ... , 19, 20);
- algoritmos: **automático**, ball tree, kd tree e brute;
- pesos: uniforme, **distância**;
- métrica: euclidiana, **manhattan**, chebyshev.

Árvore de Decisão:

- profundidade máxima: nenhuma, 3, 5, 9, **12** e 15;
- critério: gini ou **entropia**;
- divisor: **melhor** ou aleatório;
- mínimo número de divisões: **2**, 4, 6, 8;
- mínimo de amostrar por folha: **1**, 2, 3, 4, 5;
- número máximo de atributos: **nenhum**, automático, raiz quadrada ou log2.

MLP:

- função de ativação: tangente hiperbólica, relu, **identidade**, logística;
- solver: **lbfgs**, sgd, adam;
- alpha: **0.01**, 0.001, 0.0001, 0.00001.

Naive Bayes:

- variável de suavização: $10^{-9}, 10^{-8}$, ...,10^{-3}, 10^{-2}, 10^{-1}.

SVM:

- parâmetro de regularização: 0.1, 0.3, 0.5, 0.75, **1**;
- kernel: **linear**, rbf, sigmoid.

Foram comparadas as métricas dos algoritmos quando utilizado todos os atributos versus retirando os atributos Fumante e Monitoramento do consumo de calorias (que foram avaliadas como as piores características em importância pros modelos). A diferença foi irrelevante, e em alguns casos, as métricas com todos os atributos foram superiores sem retirada deles.

O último passo foi aplicar os modelos otimizados para cada algoritmo de machine learning no conjunto de testes, a fim de obter as métricas finais para o problema proposto.

Aplicando a curva ROC ao modelo K-NN foi possível analisar a performance das classes individualmente, ao longo do processamento como exibido na Figura 6, denota-se que houve uma alta taxa de acertos na predição para a maioria das classes, havendo uma perca na qualidade na classe 'Sobrepeso nível II' onde apresenta um valor AUC $= 0.89$, por outro lado a classe 'Obesidade nível III' teve uma taxa total de acertos.

No modelo *Decision Tree* mostrado na Figura 7 novamente uma taxa de acertos altas mostrando pouca variação entre as classes, e apresentando a mesma perca na classe 'Sobrepeso nível II' e 100% de acertos na classe 'Obesidade nível III'.

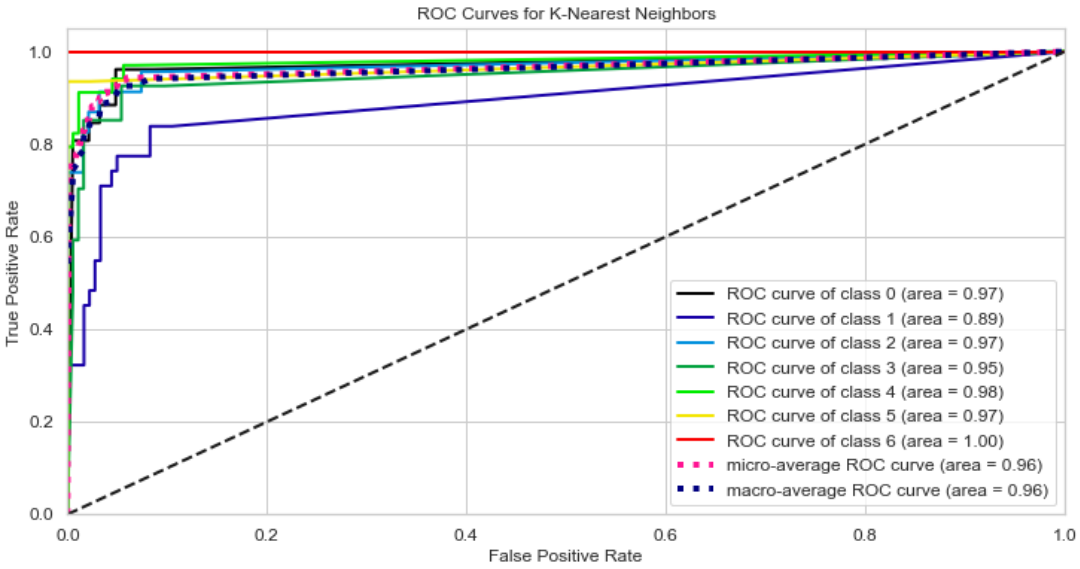

Figura 6. *Curva ROC e área AUC do algoritmo K-NN*

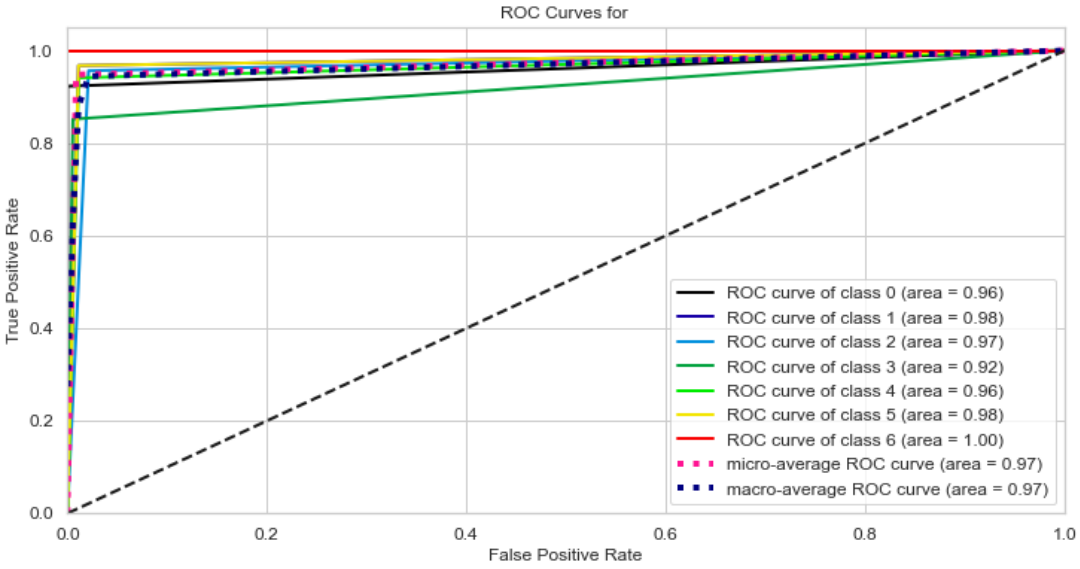

Figura 7. *Curva ROC e área AUC do algoritmo Decision Tree*

Para o modelo MLP exibido na Figura 8 a o erro existente é praticamente mínimo, o algoritmo converge para uma resolução quase perfeita, o que a princípio pode ser um bom indicativo de overfitting. Outra das possível causas para esse resultado poderia ser a trivialidade do problema. De toda maneira, neste caso o modelo executou bem a predição dos dados.

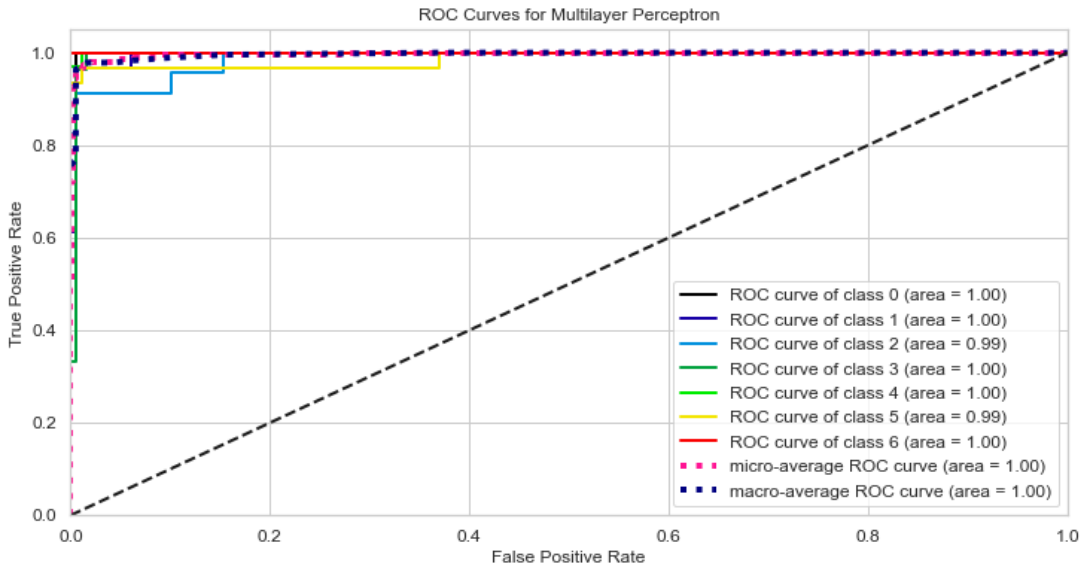

Figura 8. *Curva ROC e área AUC do algoritmo MLP*

No modelo de *Naive Bayes* apresenta os valores mais variantes na predição conforme pode ser visualizado na Figura 9, onde a classe 'Obesidade tipo I' apresenta o menor índice do valor AUC $= 0.82$, entretanto a classe 'Obesidade nível III' permanece sem erros na sua predição.

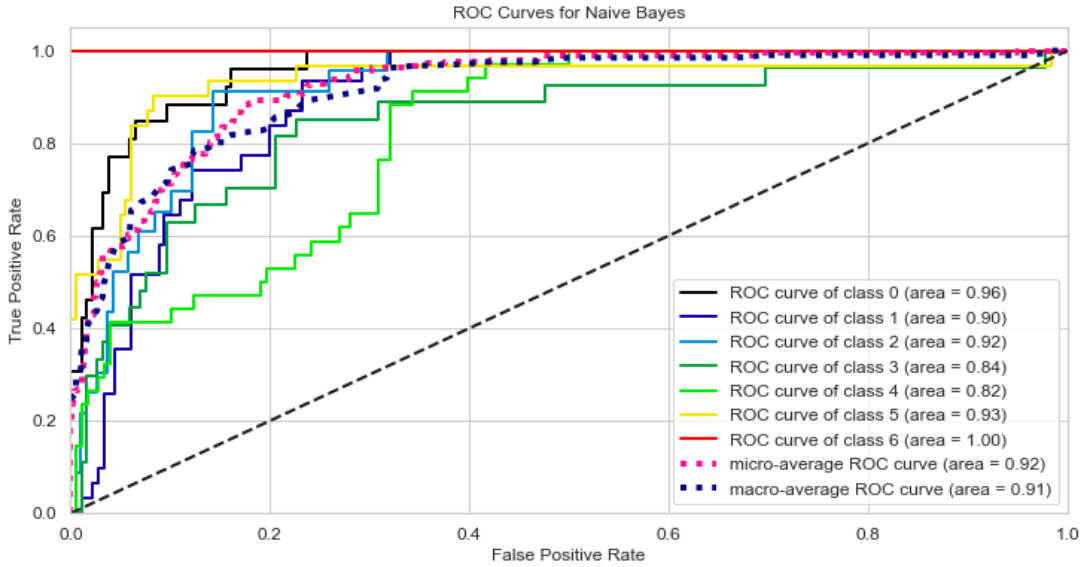

Figura 9. *Curva ROC e área AUC do algoritmo Naive Bayes*

O SVM mantém valores altos de acertos para todas classes, com AUC $\geqslant 0.97$, com pouca variabilidade nos resultados preditos como pode ser visto em 10. O comportamento se

assemelha ao da curva ROC do modelo MLP.

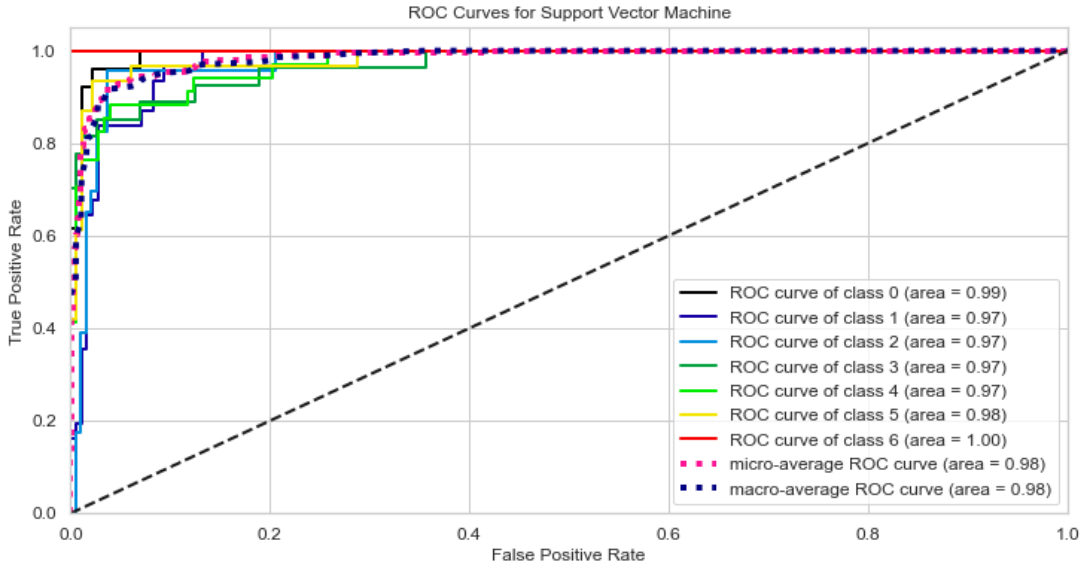

Figura 10. *Curva ROC e área AUC do algoritmo SVM*

As Figuras 11, 12, 13, 14 e 15 contém as métricas de precisão, revocação e f1-score referentes a cada algoritmo utlizado nesse trabalho

	precision	recall	f1-score	support
0	0.74	0.88	0.81	26
1	0.78	0.68	0.72	31.00
2	0.83	0.87	0.85	23.00
3	0.88	0.78	0.82	27.00
4	0.86	0.91	0.89	34.00
5	0.97	0.94	0.95	31.00
6	1.00	1.00	1.00	40.00
accuracy			0.87	212.00
macro avg	0.87	0.87	0.86	212.00
weighted avg	0.87	0.87	0.87	212.00

Figura 11. *Métricas de precisão, revocação, f1-score do algoritmo K-NN*

Por fim, as Figuras 16, 17, 18, 19 e 20 contêm as matrizes de confusão referentes a cada algoritmo utilizado neste trabalho.

	precision	recall	f1-score	support
0	1.00	0.92	0.96	26.00
1	0.94	1.00	0.97	31.00
2	0.88	0.96	0.92	23.00
3	0.96	0.85	0.90	27.00
4	0.94	0.94	0.94	34.00
5	0.94	0.97	0.95	31.00
6	1.00	1.00	1.00	40.00
accuracy			0.95	212.00
macro avg	0.95	0.95	0.95	212.00
weighted avg	0.95	0.95	0.95	212.00

Figura 12. *Métricas de precisão, revocação, f1-score do algoritmo Decision Tree*

	precision	recall	f1-score	support
0	0.96	0.96	0.96	26.00
1	0.94	0.97	0.95	31.00
2	0.95	0.87	0.91	23.00
3	0.93	0.96	0.95	27.00
4	0.97	0.97	0.97	34.00
5	1.00	0.94	0.97	31.00
6	0.95	1.00	0.98	40.00
accuracy			0.96	212.00
macro avg	0.96	0.95	0.95	212.00
weighted avg	0.96	0.96	0.96	212.00

Figura 13. *Métricas de precisão, revocação, f1-score do algoritmo MLP*

	precision	recall	f1-score	support
0	0.61	0.88	0.72	26.00
1	0.58	0.35	0.44	31.00
2	0.47	0.30	0.37	23.00
3	0.67	0.22	0.33	27.00
4	0.37	0.47	0.42	34.00
5	0.58	0.90	0.71	31.00
6	1.00	1.00	1.00	40.00
accuracy			0.62	212.00
macro avg	0.61	0.59	0.57	212.00
weighted avg	0.63	0.62	0.59	212.00

Figura 14. *Métricas de precisão, revocação, f1-score do algoritmo Naive Bayes*

	precision	recall	f1-score	support
0	0.76	0.96	0.85	26.00
1	0.83	0.65	0.73	31.00
2	0.77	0.74	0.76	23.00
3	0.75	0.78	0.76	27.00
4	0.87	0.79	0.83	34.00
5	0.85	0.94	0.89	31.00
6	1.00	1.00	1.00	40.00
accuracy			0.84	212.00
macro avg	0.83	0.84	0.83	212.00
weighted avg	0.85	0.84	0.84	212.00

Figura 15. *Métricas de precisão, revocação, f1-score do algoritmo SVM*

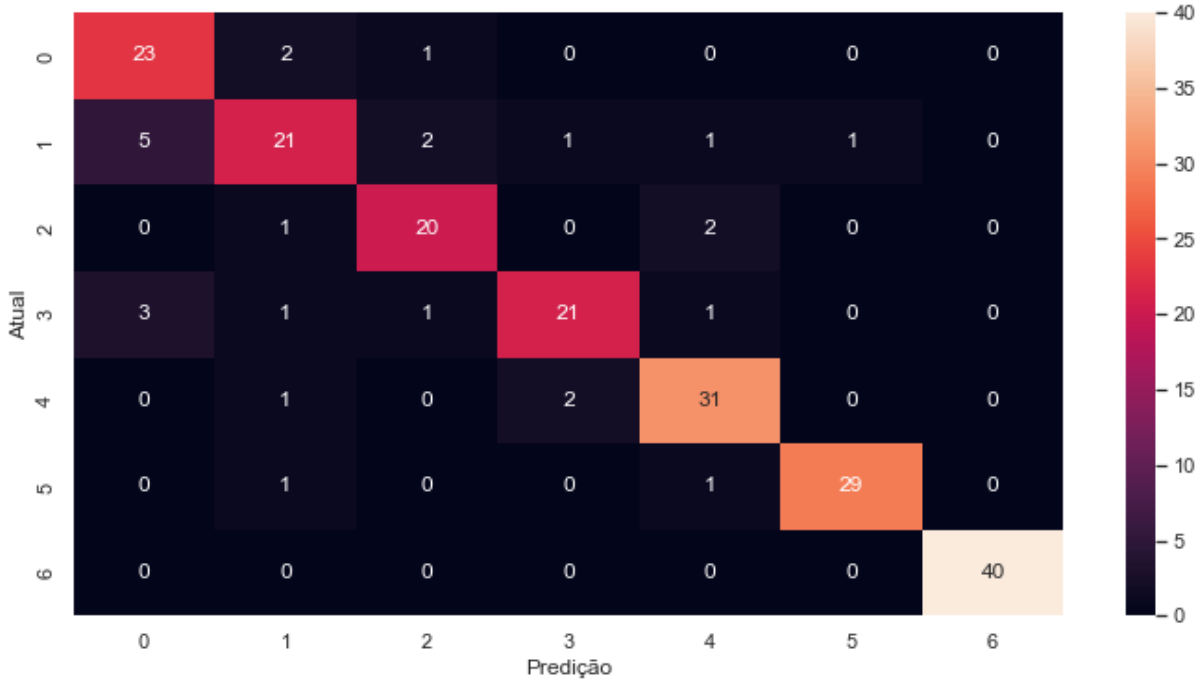

Figura 16. *Matriz confusão do algoritmo K-NN*

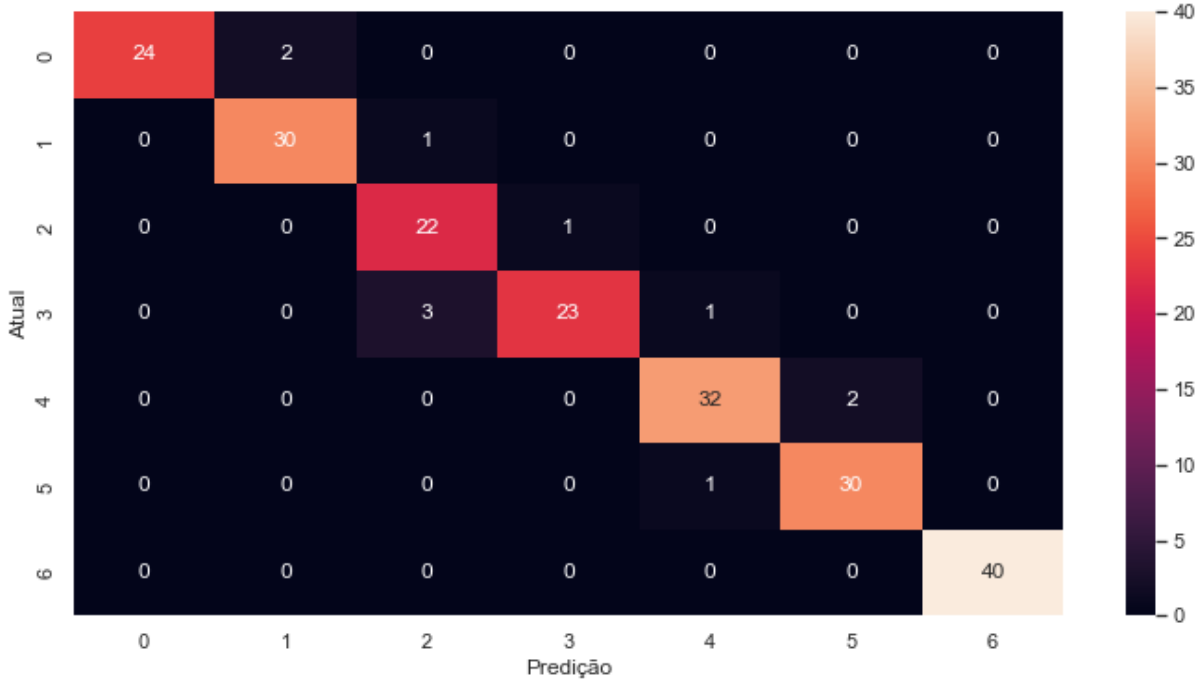

Figura 17. *Matriz confusão do algoritmo Decision Tree*

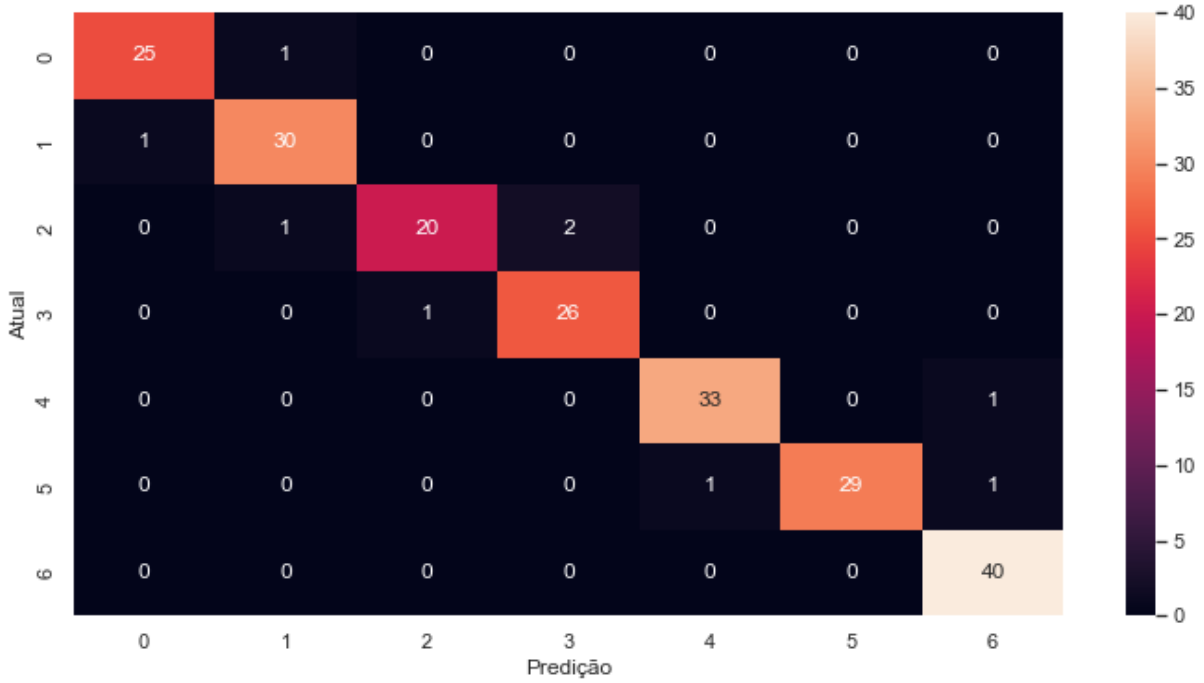

Figura 18. *Matriz confusão do algoritmo MLP*

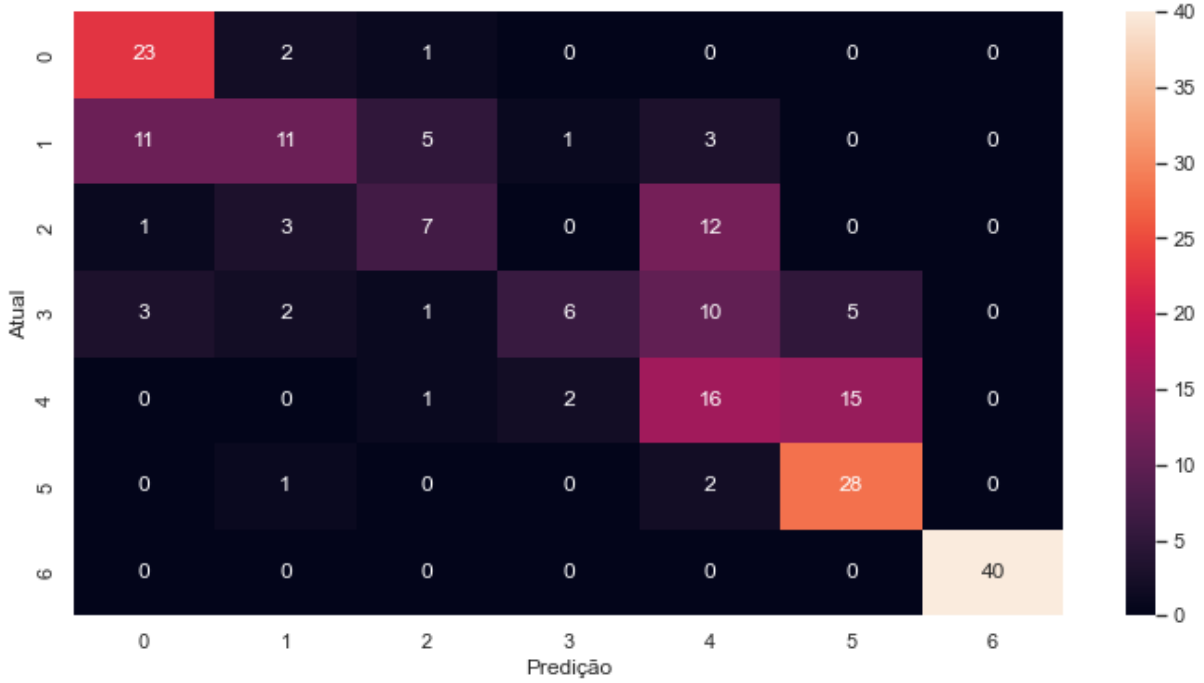

Figura 19. *Matriz confusão do algoritmo Naive Bayes*

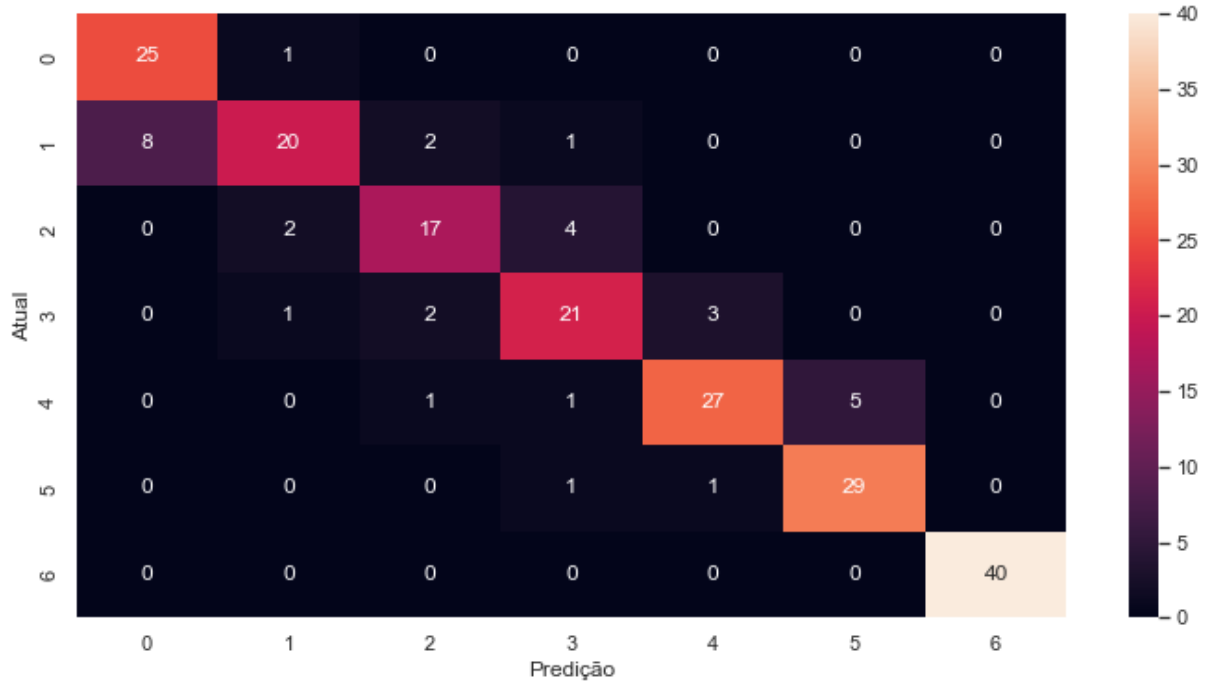

Figura 20. *Matriz confusão do algoritmo SVM*

4 Conclusões

Através deste trabalho foi possível analisar o conjunto de dados referentes à obesidade de uma amostra de indivíduos da Colômbia, Peru e México. Os algoritmos utilizados foram o K-NN, Árvore de Decisão, Multilayer Perceptron, Naive Bayes e Support Vector Machine. Exceto pelo algoritmo bayesiano, todos obtiveram um excelente resultado no conjunto de teste para todas as métricas (precisão, revocação, f1 score e acurácia). Por um lado, isso é um indício de que o conjunto de dados não é muito compatível com a natureza do algoritmo de Bayes. Em contrapartida, o emprego das técnicas de divisão em treino, validação e teste em conjunto com o k-fold estratificado e a tunagem de parâmetros via grid search permitiram que os demais algoritmos obtivessem ótimas métricas. Por fim, é também relevante apontar que os valores das métricas finais podem indicar que o problema atacado neste trabalho não é muito complexo.

Referências

1 WHO. **Obesity and overweight**. 2021. Disponível em: <https://www.who.int/en/news-room/fact-sheets/detail/obesity-and-overweight>. Acesso em: 7 janeiro 2021.

2 BLÜHER, M. Metabolically healthy obesity. **Endocrine reviews**, Oxford University Press US, v. 41, n. 3, p. 405–420, 2020.

3 PALECHOR, F. M.; MANOTAS, A. de la H. Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, peru and mexico. **Data in brief**, Elsevier, v. 25, p. 104344, 2019.

4 CHAWLA, N. V. et al. Smote: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.

5 KURSA, M. B.; RUDNICKI, W. R. et al. Feature selection with the boruta package. **J Stat Softw**, v. 36, n. 11, p. 1–13, 2010.