Métodos de Aprendizado de Máquina para Predição da Qualidade de Vinhos Brancos

1st Gabriel Pinheiro Palitot Pereira
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
gabrielppalitot@alu.ufc.br

2nd Igor Ferreira Vasconcelos
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
igorfervasc@alu.ufc.br

3rd Eliabe Bastos Dias
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
eliabebastosdias@alu.ufc.br

***Resumo*—Modelos de Classificação são utilizados em praticamente todas as áreas que envolvem análise de dados. Esses métodos são muito utilizados para classificar uma determinada amostra em uma classe, ajudando em processos como identificação. Esse trabalho busca trazer alguns métodos de classificação que são utilizados em aprendizado de máquina e usá-los para tentar predir a qualidade do vinho branco português. Os métodos utilizados para essa finalidade foram tanto métodos lineares, como o LDA, quanto não lineares, como o K-NN e Support Vector Machine (SVM). O melhor desempenho obtido com base na acurácia foi o SVM com 64% de acurácia.**

***Index Terms*—Classificação, Inteligência Artificial, Vinho, Predição, qualidade, K-NN, SVM, LDA**

I. Introdução

Aprendizado de máquina (Machine Learning) e problemas de classificação são importantes nos dias atuais e ganham cada vez mais espaço no mundo já que diversas tarefas envolvem classificar algum objeto, sinal ou em tarefas de identificação.

A classificação é mais utilizado no cotidiano do que métodos de regressão [4] já que no caso da regressão a variável de saída geralmente é uma resposta contínua e no caso da classificação termos uma saída que geralmente vem em forma categórica ou classificando para uma classe em específico, ou até mesmo a probabilidade de pertencer a uma determinada classe [3]. Portanto, para modelos de classificação é interessante o conhecimento e entendimento sobre probabilidade.

Ademais, Segundo [3], é importante citar que as formas de avaliação de desempenho para um trabalho de classificação não se baseia no uso de RMSE e R^2 para a regressão visto em [2], temos alguns outros métodos como matriz de confusão e as métricas associadas a esse método para o entendimento dos sucessos, falhas e interpretação do método de aprendizado de máquina usado para o problema de classificação.

Além disso, como é sempre importante lembrar, o pré-processamento dos dados é de suma importância para qualquer tarefa envolvendo conjuntos de dados. Métodos como centralização e dimensionamento, tratamento de assimetria e entre outros devem ser utilizados a depender do conjunto de dados. O pré-processamento auxilia em um resultado correto nos trabalhos de aprendizado de máquina.

Por fim, métodos de aprendizado de máquina como KNN, SVM e LDA são utilizados em diversas áreas de estudo ao se tratar de problemas de classificação, visto que são técnicas seguras. Esse trabalho busca prever a qualidade de vinhos brancos com a utilização desses métodos de classificação.

II. Métodos

A. Descrição do Dataset e Pré-Processamento

O dataset é o mesmo utilizado em [1]. O nome desse conjunto de dados é WineQuality, pode ser visto em [7]. Esse dataset, conforme já foi dito em [1] possui amostras de vinhos tintos e vinhos brancos, mas para esse trabalho foi utilizado as amostras de vinhos brancos. Possui cerca de 11 preditores e uma variável de saída que é a qualidade, ou seja, a classificação irá ser feita em torno da qualidade do vinho branca que é ranqueada de 1 a 10, tendo dessa forma 10 classes. Porém, conforme pode ser visto na Tabela I, não possui amostras para as notas de qualidade 1,2 e 10, portanto, foram retiradas.

As entradas incluem testes objetivos e físico-químicos, e a saída, que é a qualidade, é baseado em dados sensoriais, na qual é uma mediana das notas dadas por 3 especialistas em vinhos. Mais detalhes sobre os preditores, podem ser vistos em [1]. Além disso, não há dados faltantes para o dataset.

Já o pré-processamento dos dados, que é uma etapa importante para qualquer técnica de machine learning, visto que cada método necessita de um tratamento para os dados. A técnica básica de pré-processamento utilizada foi a centralização e o escalonamento, que se baseia na Equação 1 e faz com

Tabela I
QUANTIDADE DE AMOSTRAS POR CLASSE (QUALIDADE DO VINHO BRANCO)

Classe	Quantidade de Amostras
1	0
2	0
3	20
4	163
5	1457
6	2198
7	880
8	175
9	5
10	0

que todos os valores tenham média 0 e desvio padrão 1. É importante para casos em que possuímos escalas de dados e magnitudes diferentes [5] como é o caso desse dataset.

$$x^{'} = \frac{x - m}{sd} \tag{1}$$

Sendo que x é o valor original, x' o valor transformado, m e sd correspondem às médias e ao desvio padrão do grupo de preditores de x. De acordo com [1] esse conjunto de dados possui muita assimetria em praticamente todos os preditores, então se tornou interessante utilizar métodos como Yeo-Johnson para reduzir essa assimetria, mesmo processo foi utilizado em [2].

Por fim, os dados foram divididos em conjunto de treinamento que serve para descobrir parâmetros do modelo e em conjunto de teste que serve para ver como o modelo se saí com dados que não viu ainda. A separação foi feita em 75% para treino e 25% para teste.

B. Avaliação do Modelo: Matriz de Confusão

Um dos métodos mais utilizados para avaliação de um modelo que resolve problema de classificação é por meio da matriz de confusão. A matriz de confusão, de forma simples, faz uma comparação com os valores reais e os valores preditos na classificação e sua dimensão é CxC em que C é a quantidade de classes. No caso desse artigo, como estamos utilizando apenas as notas de 3 a 9 nesse trabalho (Visto que não há amostras para as outras notas), então teremos uma matriz de 7x7. As colunas representam o que se espera (ou seja, os resultados reais), já as linhas representam a predição. Por exemplo, se um vinho pertencer a qualidade 3 e a amostra realmente pertencer a 3, então o contador na célula 1,1 (isso porque as qualidades começam em 3, como já visto antes) vai ser 1. Além disso, a matriz de confusão nos dá algumas métricas interessantes para o avaliação do modelo, essas métricas vão ser explicadas posteriormente.

É muito comum a avaliação de matriz de confusão com dimensão 2x2, ou seja, classificação binária. É preenchida da seguinte forma: a célula 1,1 quer dizer que a predição diz que pertence àquela classe e o dado realmente pertence aquela classe, esse valor é chamado de verdadeiro positivo (VP). A célula 1,2 é que a predição diz que pertence àquela classe, mas não pertence realmente, é chamado de falso positivo (FP). A célula 2,1 a predição nos diz que não pertence àquela classe, mas a amostra pertence (chamado de falso negativo (FN)) e por fim a célula 2,2 a predição nos diz que não pertence àquela classe e realmente não pertence (chamado de Verdadeiro Negativo (VN)).

Além disso, a matriz de confusão nos dá algumas métricas interessantes para a avaliação do modelo, algumas dessas métricas e que irão ser utilizadas a depender do método são: acurácia e precisão.

A acurácia é definida como o número de amostras corretamente classificadas dadas todas as amostras. O numerador são os valores das diagonais da matriz, pois nos diz justamente quais valores foram classificados corretamente ao pertencer ou não a determinada classe. A sua fórmula é dada pela Equação 2. Para matrizes de confusão com mais de duas classes, podemos seguir o mesmo raciocínio de somar todas as diagonais (valores classificados corretamente) e dividir pelo total de amostras.

$$acuracia = \frac{VN + VP}{VP + VN + FP + FN} \tag{2}$$

A precisão são os valores que são positivos dentro os que foram classificados como positivos pelo método. Geralmente é calculado para cada classe individual e depois é feito uma média aritmética ou ponderada (pelo número de amostras), nesse trabalho foi utilizada a precisão aritmética visto que há uma quantidade diferente de amostras para cada nota de qualidade. A fórmula para a precisão está descrita na Equação 3.

$$precisao = \frac{VP}{VP + FP} \tag{3}$$

C. Análise Discriminante Linear (LDA)

Quando se fala de métodos de classificação lineares, pode-se pensar em Regressão Logística ou então em LDA que são bem famosos. Para esse trabalho foi escolhido o LDA, visto que como foi visto nas seções anteriores, possuímos mais de duas classes e o LDA lida bem com problemas de classificação que possuem mais de duas classes. Porém, primeramente é importante entender o que é o LDA ou Análise Discriminante Linear.

Para entender sobre o LDA é primeiramente necessário saber que esse método tem raízes na regra de Bayes que pode ser construída da seguinte forma, de acordo com [3]:

$$P(Y = C_l|X) = \frac{P(Y = C_l)P(X|Y = C_l)}{\sum_{l=1}^{C} P(Y = C_l)P(X|Y = C_l)} \tag{4}$$

Sendo $X = (X_1, X_2, ..., X_p)$ os preditores, C as classes e Y a saída (no nosso trabalho é a qualidade do vinho).

O termo $P(Y = C_l)$ é conhecido como probabilidade a priori, geralmente é determinado pela proporção de amostras que contém na classe C_l. Já $P(X|Y = C_l)$ é a probabilidade condicional de observamos preditores X dado que pertence a classe C_l [3]. O resultado da Equação 4 é $P(Y = C_l|X)$ que

é a probabilidade de que a observação pertence à classe C_l dado os valores dos preditores para aquela observação [4].

Outra coisa a se entender do LDA é que esse método faz algumas suposições. Uma delas é que $P(X|Y = C_l)$ segue uma distribuição normal multivariada e há um vetor médio multivariado (μ_c) que representa os valores médios dos preditores para aquela classe, e há uma matriz de covariância comum para todas as C classes (representada por Σ) [4]. Além disso, vamos usar a seguinte notação para explicação daqui para frente: $P(X|Y = C_l) = f_c$, $P(Y = C_l|X) = p_c(x)$, $P(Y = C_l) = \pi_c$

Ademais, substituindo tudo isso na Equação 4, substituindo f_c pela forma matemática da distribuição normal multivariada e tirando o log (para facilitar os cálculos), obtemos por meio de manipulação algébrica:

$$\delta_c(x) = x^T\Sigma^{-1}\mu_c - \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + log(\pi_c) \tag{5}$$

Sendo $\delta_c(x)$ a função discriminante linear para uma determinada classe c e note que $\delta_c(x) = log(P(Y = C_l|X))$. O nome linear vem do fato da função discriminante ser linear em relação a x e a sua fronteira de decisão também será linear, a fronteira de decisão ocorre quando: $\delta_c = \delta_k$ [4]. É possível ver pela Equação 5 que há termos que precisam ser estimados (μ_c, Σ e π_c) com as suas fórmulas apropriadas e os dados usados devem ser aqueles do conjunto de treino. Ao estimar esses valores, e calcular a Equação 5 para cada amostra do conjunto de teste, a amostra será colocada na classe c desde que δ_c é a maior [4], isso se deve porque $P(Y = c|X)$ é maior que $P(Y = C_1|X), ..., P(Y = C_C|X)$. Com isso é possível realizar classificação com a LDA.

D. K Vizinhos Mais Próximos (KNN)

O k-NN é um algoritmo de aprendizado supervisionado, ou seja, leva em conta a saída para o aprendizado. K-nn é não paramétrico, porém possui um hiperparâmetro, e é a partir dele que faremos todo o processamento. Neste algoritmo, temos classes definidas e usamos a quantidade de vizinhos mais próximos para classificar novos dados.

Considere um conjunto de dados onde temos os preditores X e a saída Y, onde existem N classes já bem definidas. Então recebemos um novo dado que não foi ainda classificado. Definimos, no entanto, previamente, que desejamos obter o k vizinhos mais próximos. Fazemos anteriormente o processamento e separamos entre conjunto de treino e teste. Então começamos o algoritmo:

1) Calculamos a distância do novo dado até todos os outros pontos. Seja d_i a distância entre o novo dado e o ponto i no conjunto de dados. Podemos usar, nesse momento, diversas métricas.
2) Obtemos as k distâncias mais próximas.
3) Dessas distâncias selecionadas anteriormente, contamos a quantidade de vezes que certa classe aparece e colocamos em uma tabela.
4) Toma-se, então, a classe do novo dado como a classe que mais apareceu entre as k distâncias, ou seja, a classe que maximiza a contagem.
5) Para medir a precisão do modelo, usamos diferentes métricas como precisão, acurácia, recall e f1-score. A partir dessas métricas, sabemos qual k é o melhor para ser usado.

A validação cruzada é importante porque o modelo é sensível ao tamanho do conjunto de dados e à escolha do número de vizinhos. Ao dividir o conjunto de dados em vários folds, a validação cruzada permite que o modelo seja avaliado em diferentes configurações, o que ajuda a reduzir o risco de subestimar ou superestimar o desempenho do modelo.

O algoritmo acima possui uma equação que o define:

$$P(Y = j \mid X = x_0) = 1 - \frac{1}{K}\sum_{i \in \mathcal{N}_0} I(y_i = j)$$

Perceba que nosso método considera a classificação a priori a fim de definir a classe a posteriori. A soma conta o número de vizinhos mais próximos cuja variável de saída é igual a j. A soma conta o número de vizinhos mais próximos cuja variável de saída é igual a j. A função indicadora $I(y_i = j)$ retorna 1 se $y_i = j$ e 0 caso contrário. A parte $\frac{1}{K}\sum_{i \in \mathcal{N}_0} I(y_i = j)$ normaliza o número de vizinhos mais próximos que têm $y_i = j$ pelo total de vizinhos mais próximos K.

Também não precisamos fazer o algoritmo kNN para todos os k que vão desde 1 até a quantidade de amostra N, pois o algoritmo depois de raiz de N não apresenta mudanças consideráveis. Se considerarmos um K da mesma ordem de grandeza das quantidade de amostras, o modelo não teria muito critério para colocar em uma classe ou em outra, visto que agora o modelo terá que escolher o mínimo dentro um conjunto de dados equiparável a própria base de dados.

Também é necessário não comparar os modelos com vizinhos próximos de um, pois a pouca quantidade de vizinhos pode promover um overfitting dos dados, pois o modelo vai sempre escolher que está imediatamente mais próximo dele. A quantidade de vizinhos próximo de um varia de acordo com o banco de dados de acordo com a distribuição dos dados e da correlação que eles podem possuir.

E. Máquina de Vetor de Suporte (SVM)

A Máquina de Vetor de Suporte é um método classificador avançado, não linear e multiclasse. Que utiliza como base os conceitos de hiperplanos e do classificador binário e linear Classificador de Vetor de Suporte (SVC). Para explica-lo é necessário explicar esses 2 métodos antes, começando com os hiperplanos:

Em um espaço de p-dimensões, um hiperplano é um subespaço afim de dimensões p-1 que satisfaz a equação 6. Em outras palavras, um espaço de 2 dimensões possui um subespaço reto de 1 dimensão, um espaço de 3 dimensões possui um subespaço plano de 2 dimensões. Dessa forma, conseguimos mostrar hiperplanos em gráficos, mas, representá-los em níveis p > 3 é praticamente impossível.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0 \tag{6}$$

Se esta é a fórmula do hiper plano, queremos calcular os betas e X adequados de forma que valores em classes diferentes

tenham valores diferente de zero, então se Y > 0, o elemento pertence a uma classe e se Y < 0, o elemento pertence a outra

No entanto, esse método busca classificar 100% dos dados, dessa forma, novos dados podem alterar drasticamente a função do hiperplano, em especial, dados ruidosos podem gerar uma equação de hiperplano sem solução. Dito isso, usamos funções de maximização e penalização para adicionar robustez ao modelo, que é o caso do SVC.

O SVC utiliza os hiperplanos para gerar um espaço que separa uma parcela significativa dos dados, de forma a delimitar a fronteira que separe as classes de forma satisfatória, mas não total.

[4] É utilizado caso não exista solução para a equação do hiperplano ou no caso que a separação total não seja desejada. Visto que a separação total pode gerar overfitting e cada novo dado adicionado pode alterar drasticamente a equação do hiperplano.

Assim, semelhante a métodos de regressão penalizada, visto em [2] o classificador se ajusta menos ao conjunto de treino, evitando o overfitting e se torna um modelo mais robusto. Dessa forma, se ajusta melhor a uma variedade maior de dados, especificamente, ao conjunto de testes.

Esse método se resume a um problema de maximização, referente as equações 7, 8, 9, 10.

$$\max_{\beta_0,\beta_1,\ldots,\beta_p,M} M \tag{7}$$

$$\sum_{j=1}^{p}(\beta_j)^2 = 1 \tag{8}$$

$$y_i(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p) \geq M(1-\epsilon_i), \quad \forall i = 1,\ldots,n \tag{9}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n}(\epsilon_i) \leq C \tag{10}$$

Onde C é um hiper parâmetro, M é a largura da margem que buscamos maximizar, $\epsilon_1, ..., \epsilon_n$ são as variáveis de folga, que permitem que observações individuais estejam do lado errado do hiperplano. Quanto maior o valor de C mais tolerante é o modelo, ou seja, semelhante ao λ da Regressão Ridge visto em [2], o C controla o equilíbrio viés-variância.

Nos conjuntos de dados que possuem mais de 2 classes e nos casos que as margens lineares não são suficientes precisamos usar a SVM. Que trata de reduzir a linearidade ao expandir a dimensionalidade dos dados utilizando um processo de kernalização.

A função que define a SVM é definida pela equação 11

$$f(x) = \beta_0 + \sum_{i \varepsilon S} \alpha_i(K(x, x_{i'})) \tag{11}$$

Em que S é o conjunto dos pontos de vetores de suporte. Para descobrir o valor de β_0 e α_i precisamos somente do produto interno das observações $< x, x_i >$, ou seja as observações dos pontos de suporte.

A forma como vamos utilizar esses produtos internos e o formato do hiper plano é determinada pelas funções de kernel K(x, x_i). As equações12 e 13 tratam respectivamente dos kernels de função de base radial (rbf) e polinomial.

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p}(x_{ij} - x_{i'j})^2) \tag{12}$$

Em que gamma é outro hiper parâmetro de tolerância.

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p}(x_{ij}x_{i'j})^d \tag{13}$$

Em que d é o número de dimensões do polinômio que define o hiper plano.

Existem outras funções, mas como sabemos que o nosso conjunto de dados é complexo, como vimos em [1] e [2], esses são os kernels mais prováveis de serem utilizados.

F. Curva característica de operação do receptor (ROC)

A curva característica de operação do receptor (ROC) é uma ferramenta gráfica usada para avaliar o desempenho de um classificador. Ela plota a taxa de verdadeiros positivos (TPR) contra a taxa de falsos positivos (FPR), onde:

1) TPR é a proporção dos exemplos positivos que são corretamente classificados como positivos.
2) FPR é a proporção dos exemplos negativos que são incorretamente classificados como positivos.

Uma curva ROC ideal é uma linha diagonal, que indica que o classificador é perfeito. No entanto, na prática, as curvas ROC são sempre curvas, com uma forma que depende das características do classificador e do conjunto de dados.

A área sob a curva ROC (AUC) é uma medida numérica do desempenho do classificador. Uma AUC de 1 indica que o classificador é perfeito, enquanto uma AUC de 0,5 indica que o classificador é aleatório.

A curva ROC é composta por três pontos principais:

1) Ponto (0,0): Este ponto indica que o classificador não classifica nenhum exemplo como positivo ou negativo.
2) Ponto (1,0): Este ponto indica que o classificador classifica todos os exemplos como positivos ou negativos, com precisão perfeita.
3) Ponto de inflexão: Este ponto indica a transição entre a região de melhor desempenho do classificador e a região de pior desempenho.

A curva ROC pode ser interpretada de várias maneiras. Uma forma comum é comparar a curva ROC de dois ou mais classificadores diferentes. O classificador com a curva ROC mais alta terá o melhor desempenho.

A curva ROC também pode ser usada para escolher o ponto de corte ideal para o classificador. O ponto de corte é o valor que determina se um exemplo será classificado como positivo ou negativo.

III. Resultados

A. Análise Discriminante Linear (LDA)

Primeiramente, vai ser analisado os resultados de um método linear, que no caso utilizado nesse trabalho foi o LDA ou Análise Discriminante Linear.

Como foi dito no pré-processamento, foi dividido o conjunto de dados em conjunto de treino e conjunto de teste em 75% e 25%, respectivamente. Após isso, foi aplicado o conjunto de treino para realizar o treinamento, ou seja, descobrir os parâmetros do LDA, que são a probabilidade a priori de cada classe (π_c), a média de cada classe (mu_c) e matriz de covariância (Σ) que é comum para as classes. Por fim, é colocado os parâmetros descobertos no treinamento na Equação 5 em conjunto com o vetor de testes x para ver a predição para cada amostra. Na Tabela II pode-se ver os resultados obtidos de acurácia e precisão. Lembrando que foram usadas as classes de notas de qualidade de 3 a 9.

Tabela II
Resultados do modelo LDA

Modelo	Acurácia	Precisão
LDA	0.538	0.603

A acurácia e precisão vistas na Tabela II são interessantes de serem avaliadas. É possível notar que a acurácia não foi um resultado bom, já que o seu valor vai de 0 a 100% e foi conseguido um valor de 53.8% , o que mostra que dentre todas as predições, apenas aproximadamente metade delas foram feitas de forma correta, isso pode ser devido ao fato dos dados não conseguirem ser separados por meio de uma fronteira de decisão linear, por exemplo, ou porque o LDA não é um método adequado para classificação da qualidade dos vinhos. Sobre a precisão, vê-se que atingiu um resultado de cerca de 60% o que mostra que dentre as amostras que foram classificadas como positivas, cerca de 60% são verdadeiramente positivas, o que é um resultado mediano, pois pode mostrar que cerca de 40% das amostras estão sendo classificados como não pertencentes a uma determinada nota de vinho branco, por exemplo. Para uma visualização mais completa do desempenho do modelo, pode-se ver a matriz de confusão na Figura 4.

De acordo com a Matriz de confusão do método LDA (Figura 4) se vê que as classes de notas de qualidade de vinho 5 e 6 são as mais acertadas (olhar para a diagonal da classe 5 e 6). Além disso, pode-se ver que as classes das extremidades não possuem muitos acertos. Algumas classes o modelo não acertou nenhuma classificação, como é o caso da classe de nota 8. Então, por meio dessa matriz de confusão, pode-se ver que o modelo conseguiu acertar relativamente bem os de classe 6 (cerca de 75%) enquanto as classes mais extremas como 3,8 e 9 não conseguiu acertar praticamente nenhuma classificação.

B. K Vizinhos Mais Próximos (KNN)

Por meio desse método, foi possível obter uma acurácia de 54,0%. Essa acurácia, no entanto, só foi obtida porque

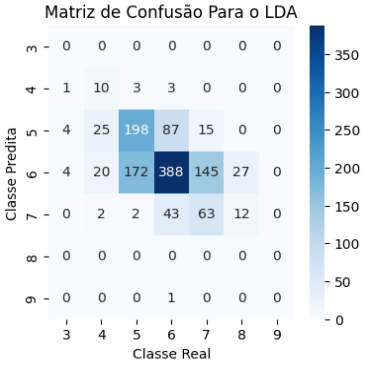

Figura 1. Matriz de Confusão LDA - Vinhos Brancos

descartamos os 9 primeiros vizinhos. Visto que eles ofereciam uma alta performance, demonstrando o overfitting explicado na teoria. Escolhemos a quantidade máxima de setenta vizinhos, pois temos 4898 e 70 é o menor inteiro maior que raiz da quantidade de amostras.

Para compararar, foi feito o kNN tanto para de um a setenta quanto para dez a setenta. Para comparar, vejamos o gráfico que acompanha as métricas de acordo com a quantidade de vizinhos. Foi utilizado a acurácia como norteador da decisão do melhor k. No caso de um a setenta, todos os k de um a dez possuem acurácia melhor do que qualquer outro k. No entanto, quando retiramos esses nove, conseguimos uma maior regularidade e, portanto, um modelo mais adequado para classificar novos dados.

Perceba que a precisão e a acurácia possuem valores muito próximos, portanto, não são distiguidos em cor.

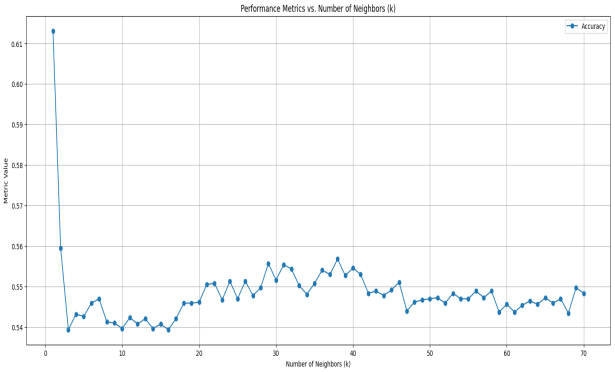

Figura 2. Métrica para o KNN, testando de a 1 a 70 vizinhos para o vinho branco

Executando os scripts, encontra-se, portanto que trinta e oito é a melhor quantidade de vizinhos, com 54,0% de acurácia. Com certeza, não é modelo com maior acurácia, mas consegue ter uma maior que os lineares. A matriz de confusão para tal método pode ser descrito como na figura 4.

C. Máquina de Vetor de Suporte (SVM)

De longe, foi o método mais bem-sucedido, a acurácia foi de 64,33%. Devido aos ajustes finos feitos ao modelo por meio do

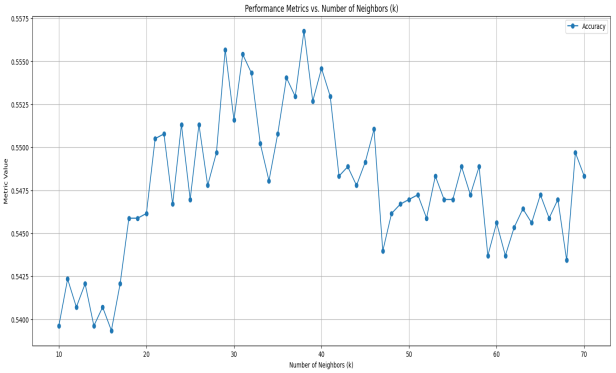

Figura 3. Métrica para o KNN, testando de a 10 a 70 vizinhos para o vinho branco

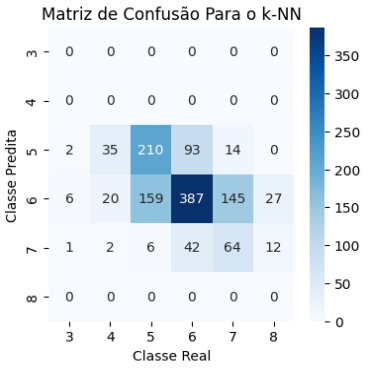

Figura 4. Matriz de confusão para o 38 vizinhos

método da busca em grade com validação cruzada. Que trata de realizar a validação cruzada com todas as possibilidades de kernel, valores para C, gamma e d. Baseando-se na natureza do conjunto de dados, como fora dito na seção II, elencamos os kernels função de base radial e polinomial, e realizamos testes sucessivos para descobrir o C mais adequado, o grau do polinômio (d) mais adequado e o γ mais adequado para eles.

Com os resultados, optamos pelo rbf. Com C = 2680 e γ = 0.7777, o valor de C foi elevado, pois como nosso conjunto de dados não é balanceado, precisamos que o classificador possua um forte bias, de forma que ele consiga separar os vinhos com menor (nota 4 para baixo) e maior (nota 8 para cima) classificação dos vinhos de classificação média (nota 5 e 6).

Isso permitiu uma matriz de confusão que não é tão densa no centro, classificando melhor os vinhos nos extremos, como visto na figura 5.

D. Comparação entre os métodos

SVM é considerado um classificador mais poderoso do que kNN e LDA. Isso ocorre porque SVM pode encontrar uma fronteira de decisão linear ou não linear que separa as classes de dados com precisão. kNN, por outro lado, é um classificador baseado em distância que classifica uma instância com base em suas k instâncias mais próximas. LDA é um classificador de análise discriminante linear que assume que

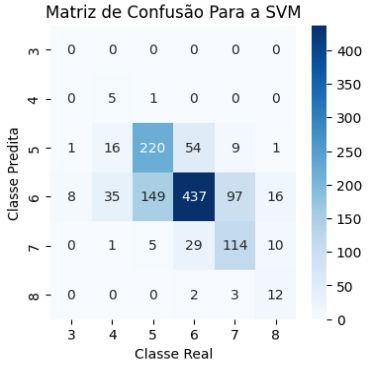

Figura 5. Matriz de Confusão da SVM

as classes de dados têm distribuições normais e com uma fronteira de decisão linear. Portanto, o kNN consegue ser mais poderoso do que LDA, visto que consegue identificar fronteiras de maneira não linear, o que LDA não consegue fazer. Isso é perfeitamente demonstrado no gráfico 6. Porém, o SVM consegue se sobressair sobre todos os outros. Percebe-se que o SVM ficou bem próximo do KNN o que demonstra que para essa amostras de dados de vinhos os classificadores não-lineares são mais interessantes, inclusive pode-se afirmar isso tendo em vista os resultados obtidos e o gráfico da Figura 6.

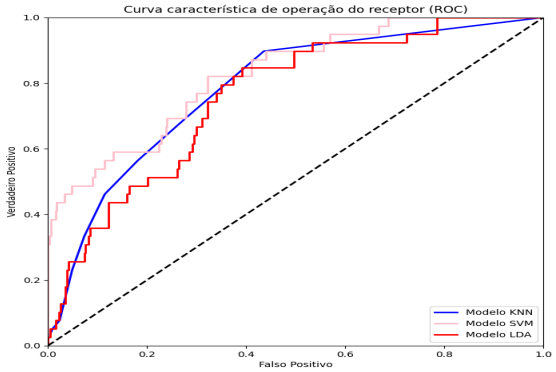

Figura 6. Gráfico ROC que compara modelos

REFERÊNCIAS

[1] Pinheiro, Gabriel P. P., Vasconcelos, Igor F., Dias, Eliabe B., (2023). Estatística Descritiva Como Ferramenta Para Análise Exploratória de Qualidade de Vinho Português.

[2] Pinheiro, Gabriel P. P., Vasconcelos, Igor F., Dias, Eliabe B., (2023). Métodos para Predição da Concentração de Álcool em Vinhos Brancos.

[3] Kuhn, Max, and Kjell Johnson. Applied Predictive Modeling. Springer, 2013.

[4] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in Python, 1st ed., New York :Springer, 2023

[5] Deisenroth, Marc Peter, Faisal, A. Aldo and Ong, Cheng Soon. Mathematics for Machine Learning. Cambridge University Press, 2020, pp 335,336,337

[6] T. Hastie, R. Tibshirani, J. Friedman Elements of Statistical Learning, Springer, 2017.

[7] Cortez, P. et al. “Modeling wine preferences by data mining from physicochemical properties.” Decis. Support Syst. 47 (2009): 547-553.