Aplicação de Modelos de Classificação sobre Dados de aplicações de Bolsa na universidade de Melbourne

1st Cícero Oliveira da Silva Júnior
Departamento de Engenharia Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
junioroliver@alu.ufc.br

2nd David Lima dos Santos
Departamento de Engenharia Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
david.santos@alu.ufc.br

3rd Gabriel Moreira de Andrade
Departamento de Engenharia Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
gabrielm03@alu.ufc.br

4th Igor Pereira Gouveia
Departamento de Engenharia Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
igorpereiragv@alu.ufc.br

***Abstract*—O presente trabalho apresenta a aplicação de diferentes modelos de classificação sobre um mesmo conjunto de dados a fim de comparar seus resultados. Para tal, utiliza-se um conjunto de dados sobre aplicações para bolsa com 1882 váriaveis preditoras e 8708 observações. Os modelos utilizados são: regressão logística, k-vizinhos mais próximos (do inglês *k-nearest neighbours*), máquina de vetores de suporte. Utilizam-se ainda curva ROC e matriz de confusão para análise de performance. São obtidas e analisadas métricas para os resultados de implementações dos modelos citados.**

***Index Terms*—classification, logistic regression, k-nearest neighbours, support vector machine**

I. Introdução

A classificação em Machine Learning é uma tarefa essencial que busca categorizar dados em diferentes classes com base em padrões aprendidos a partir de um conjunto de treinamento. Esse processo envolve a construção de modelos capazes de generalizar e realizar predições em novos dados.

Os modelos de classificação podem ser categorizados em lineares e não lineares. Modelos lineares, como a Regressão Logística e a Análise Discriminante Linear (LDA), assumem que os dados podem ser separados por uma fronteira de decisão linear. Já os modelos não lineares, como o K-Nearest Neighbors (KNN), a Análise Discriminante Quadrática (QDA), as Máquinas de Vetores de Suporte com *Kernels* e as Redes Neurais, são mais flexíveis e conseguem capturar relações complexas entre os dados, tornando-os ideais para problemas onde uma separação linear não é possível.

No processo de modelagem preditiva para classificação, é fundamental analisar e comparar diferentes modelos a fim de identificar aquele que melhor se adapta ao problema. Além disso, é importante ainda considerar técnicas de validação, como validação cruzada de k-dobras e transformações dos dados para reduzir assimetria a fim de assegurar uma maior robustez dos resultados e melhorar a performance do modelo.

II. Metodologia

A. Análise exploratória e pré-processamento dos dados

O conjunto de dados da universidade de Melbourne, na Austrália, [1] contém D = 1882 variáveis preditoras e N = 8708 observações. É fornecido um subconjunto reduzido dos preditores originais para utilização nos modelos de classificação, com $D_{Reduzido}$ = 252 preditores, obtidos após filtragem dos preditores originais devido à alta correlação. Esse conjunto reduzido é filtrado novamente nesse trabalho, pois, entre os 252 preditores, ainda há alguns com alta colinearidade.

O conjunto de dados tem dados sobre diversas características de aplicações para pedido de bolsa para a Universidade de Melbourne, como cargo do indivíduo, linguagem materna, data de submissão da aplicação, dentre outros. A variável de saída é a classificação do indivíduo no processo de bolsa: bem-sucedido ou mal-sucedido.

Dentre as técnicas de pré-processamento, optou-se por aplicar a transformação de Box-Cox, método estatístico apresentando em 1964 por Box e Cox, utilizado neste trabalho para reduzir a assimetria da distribuição dos dados, já apresentado no trabalho anterior [7]. Também procedeu-se a centralização e normalização dos dados a fim de melhorar a estabilidade numérica dos mesmos.

B. Métricas de performance

Diferente dos métodos de regressão, os métodos de classificação podem fornecer uma predição binária — isto é, 0 ou 1 — além da probabilidade de pertencimento de classe — um valor probabilístico entre 0 e 1 do quão uma determinada amostra pertence a uma classe —, a qual é contínua [8]. Com o objetivo de servir de parâmetros de avaliação para os métodos que serão aplicados no conjunto de dados, usou-se a matriz de confusão, a acurácia, e a curva ROC.

A matriz de confusão é um método avaliativo que relaciona em uma matriz a relação dos valores reais com os valores

preditos. O eixo x representa o eixo x representa as classes reais e o eixo y representa as classes previstas pelo modelo.

TABLE I
MATRIZ DE CONFUSÃO

		Valores Reais	
		0	1
Valores preditos	0	VN	FN
	1	FP	VP

onde

- VP (Verdadeiros positivos): Amostras que foram corretamente classificados como positivos pelo modelo.
- VN (Verdadeiros negativos): Amostras que foram corretamente classificados como negativos pelo modelo.
- FP (Falsos Positivos): Amostras que foram incorretamente classificados como positivos pelo modelo.
- FN (Falsos Negativos): Amostras que foram incorretamente classificados como negativos pelo modelo.

Por meio desta, averiguou-se a quantidade de instâncias bem preditas pela diagonal de VP e VN [9]. Além disso, a partir dessas categorias vistas nos quadrantes, é possível usá-los para outros parâmetros de avaliação.

A acurácia mede a proporção de predições corretas feitas pelo modelo em relação ao total de predições [10]. Ela pode ser encontrada a partir dos valores encontrados na matriz de confusão:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}$$

Entretanto, ela, por si só, não consegue dar uma avaliação completa sobre o funcionamento do modelo. Tem-se também, a sensibilidade (ou *recall*) e a especificidade.

$$\text{Sensibilidade} = \frac{VP}{VP + FN}$$

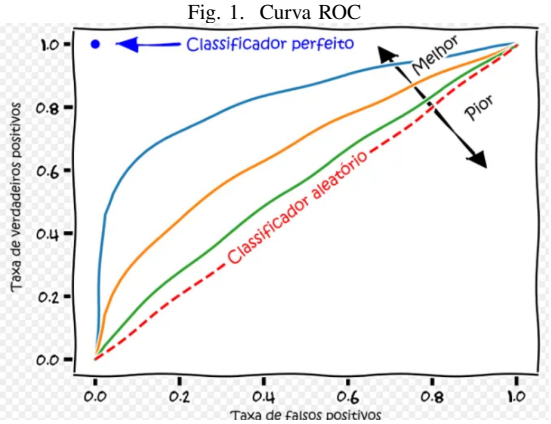
$$\text{Especificidade} = \frac{VN}{VN + FP}$$

O *Recall* representa a proporção de dos resultados positivos corretos em relação a todos os resultados positivos reais. A especificidade é a razão entre as amostras preditas corretamente como negativas por aquelas que realmente não são pertecentes a classe [10].

Assim como existe o *Tradeof Bias-Variance*, existe o *Tradeof* do recall com a especificidade. Isto é, a melhora de um significa a piora do outro. No entanto, a métrica priorizada depende sempre do contexto de aplicação, para alguns casos, pode ser melhor ter uma melhor especificidade, em outros uma melhor sensibilidade. Uma melhor sensibilidade implica em um modelo que tem uma ótima capacidade em minimizar o número de falsos negativos, uma vez que ele é melhor em não perder uma amostra verdadeira-positiva [8].

A Curva de característica de operação do receptor (Receiver Operating Characteristic, em inglês) é um gráfico que mostra a taxa de verdadeiro positivo (*recall*) pela a taxa de falso positivo para diferentes limiares. Essa taxa de falso positivo indica a razão de instâncias preditas incorretamente como negativas [11]. Um modelo com essa taxa alta indica que ele produz resultados indicados como ”alarme falso” (*False alarm*, em inglês). Dessa forma, ela é dada por $1 - especificidade$ [8]. Enquanto o recall busca detectar instâncias positivas, a especificidade, ou taxa de verdadeiro negativo, quer que o modelo tenha bom desempenho em detectar corretamente instâncias negativas. Logo, a figura 1 representa, em forma gráfica, o *tradeof* entre o *recall* e a especificidade.

Fig. 1. Curva ROC



A área sob a curva (*Area under curve*, em inglês) é um ótimo fator de análise de um modelo pois ela nos permite visualizar a capacidade do modelo de distinguir entre duas classes (positivas e negativas) [11]. Um valor próximo a 0 significa que o modelo não consegue distinguir bem entre as duas classes. Um valor de AUC igual a 0.5, representado pela linha vermelha significa que o modelo está predizendo valores aleatórios para as amostras. O valor mais próximo de 1 significa que o modelo está conseguindo distinguir entre as classes corretamente.

C. Regressão Logística

A regressão logística é uma técnica estatística utilizada para modelar a probabilidade de ocorrência de um evento com base em variáveis independentes contínuas e/ou binárias. O principal objetivo é estimar a probabilidade de uma variável dependente categórica assumir determinados valores, utilizando a função logística, definida como:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

Em que z é uma combinação linear das variáveis independentes:

$$z = \mathbf{w} \cdot \mathbf{x} + b \tag{2}$$

Nesta equação, $\mathbf{w}$ representa o vetor de pesos, $\mathbf{x}$ o vetor de características e b o viés. A função logística garante que os valores permaneçam no intervalo $[0, 1]$, permitindo a interpretação da saída $g(z)$ como uma probabilidade.

A regressão logística pode ser aplicada em dois cenários principais: classificação binária e classificação multiclasse. O enfoque será em explicar o cenário da classificação binária, devido à características da saída que se busca prever. Em ambos os casos, o modelo define fronteiras de decisão que separam as diferentes classes no espaço das variáveis independentes.

Na classificação binária, o objetivo é classificar as amostras em duas categorias. A probabilidade de uma amostra pertencer à classe 1 ($Y = 1$) é modelada por:

$$P(Y = 1|X = x) = \frac{1}{1 + e^{-(\mathbf{w}\cdot\mathbf{x}+b)}} \tag{3}$$

A decisão sobre a classe da amostra é feita comparando-se essa probabilidade a um limiar θ, comumente 0,5. Contudo, a depender da aplicação em que este modelo está sendo usado pode-se mudar esse limiar de modo a ter mais controle da ocorrência de falsos positivos ou negativos.

$$\begin{cases} \text{P}(Y = 1 \mid X = x) \geq \theta, & \text{classe 1} \\ \text{caso contrário} & \text{classe 2} \end{cases}$$

A fronteira de decisão é definida pelo conjunto de pontos onde $P(y = 1|\mathbf{x}) = \theta$, resultando em:

$$\mathbf{w} \cdot \mathbf{x} + b = -\ln\left(\frac{1-\theta}{\theta}\right) \tag{4}$$

Essa equação representa um hiperplano no espaço das variáveis independentes que separa as duas classes. Em casos com duas variáveis preditoras (x_1 e x_2), a fronteira de decisão será uma linha reta no plano x_1-x_2.

A estimativa dos parâmetros é feita pelo método da máxima verossimilhança, que busca maximizar a probabilidade dos dados observados:

$$L(\mathbf{w}, b) = \prod_{i=1}^{N} P(y_i|\mathbf{x}_i)^{y_i}(1 - P(y_i|\mathbf{x}_i))^{1-y_i} \tag{5}$$

Maximizar essa função equivale a minimizar a função de perda de entropia cruzada:

$$J(\mathbf{w}, b) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right] \tag{6}$$

Em que $\hat{y}_i$ representa a probabilidade prevista para a observação i. Os pesos e o viés são ajustados usando o gradiente descendente:

$$\mathbf{w} = \mathbf{w} - \alpha\frac{\partial J}{\partial \mathbf{w}} \tag{7}$$

$$\mathbf{w} = \mathbf{w} - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{1 + e^{-(\mathbf{w}^T X^{(i)}+b)}} - y^{(i)}\right)X^{(i)} \tag{8}$$

$$b = b - \alpha\frac{\partial J}{\partial b} \tag{9}$$

$$b = b - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{1 + e^{-(\mathbf{w}^T X^{(i)}+b)}} - y^{(i)}\right) \tag{10}$$

Em que;

- α é a taxa de aprendizado.
- $X^{(i)}$ é o vetor de entrada da i-ésima amostra da matriz de dados.
- $y^{(i)}$ = rótulo da i-ésima amostra do vetor de saída.

D. K-Vizinhos Mais Próximos

O **K-Vizinhos Mais Próximos** (do inglês *K-Nearest Neighbours*, abreviado KNN) é um método de classificação não paramétrico, ou seja, ele não assume uma estrutura específica de dados para lidar com o problema. O princípio fundamental do KNN é que objetos semelhantes tendem a estar próximos no espaço relativo às características.

Assim, o objetivo é encontrar o melhor K (número de vizinhos a serem analisados) para cada objeto. Para isso, é preciso antes determinar a distância. Com isso, consegue-se selecionar os k vizinhos mais próximos e estimar a probabilidade condicional.

Para cada classe j como a fração de pontos em N_0 (os K vizinhos mais próximos) pertencem a classe j:

$$P(Y = j|X = x_0) = \frac{1}{k}\sum_{i\in N_0} I(y_i = j) \tag{11}$$

Em que $I(y_i = j)$ é uma variável indicadora que assume o valor 1 se $y_i = j$ e 0 caso contrário.

Após isso, pode-se aplicar a regra de decisão para que x_0 seja adicionado à classe com maior probabilidade.

Há varias maneiras de calcular a distância calculada, essas são as 4 principais:

1) **Distância Euclidiana**:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2} \tag{12}$$

A distância euclidiana é mais comum e mais usada no KNN. Funciona simplesmente utilizando a distância reta mais curta entre 2 pontos. Costuma ir melhor com dados que possui dimensões semelhantes.

2) **Distância de Manhattan**:

$$d(x, x_i) = \sum_{j=1}^{n}|x_j - x_{ij}| \tag{13}$$

A distância de Manhattan mede a soma das diferenças absolutas entre as coordenadas. Isso faz com que ela seja menos sensível a outliers.

3) **Distância de Chebyshev**:

$$d(x, x_i) = \max_j |x_j - x_{ij}| \tag{14}$$

A distância de Chebyshev, considera a maior diferença em qualquer dimensão.

4) **Distância de Minkowski**:

$$d(x, x_i) = \left(\sum_{j=1}^{n}|x_j - x_{ij}|^p\right)^{\frac{1}{p}} \tag{15}$$

Em que p é um parâmetro ajustável que define o grau da distância.
A distância de Minkowski basicamente generaliza as distâncias Euclidiana e de Manhattan usando p. Assim, a depender do valor do p, ela consegue simular essas distâncias. [5]

E. Máquina de Vetores de Suporte

A **Máquina de Vetores de Suporte** (do ingles *Support Vector Machine*, abreviado *SVM*) é um algoritmo de aprendizado de máquina supervisionado que classifica dados identificando o hiperplano ótimo que maximiza a margem entre as classes em um espaço N-dimensional. A margem é definida como a distância entre o hiperplano e os pontos de dados mais próximos de cada classe, conhecidos como *vetores de suporte*. [3]

Para encontrar esse hiperplano ótimo, a Máquina de Vetores de Suporte busca maximizar a margem enquanto classifica corretamente os dados de treinamento. Esse problema pode ser formulado como uma tarefa de otimização com restrições:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \tag{16}$$

Sujeito a:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad \forall i \tag{17}$$

Em que $\mathbf{w}$ é o vetor de pesos perpendicular ao hiperplano, b é o termo de viés, $\mathbf{x}_i$ representa as características de entrada e y_i são os rótulos das classes ($+1$ ou -1).

Para resolver esse problema, utilizam-se os multiplicadores de Lagrange, construindo o Lagrangiano:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1] \tag{18}$$

onde $\alpha_i \geq 0$ são os multiplicadores de Lagrange. A solução ótima é obtida ao maximizar L em relação a α_i e minimizar em relação a $\mathbf{w}$ e b. De modo a se reduzir ao problema dual:

$$\max_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^\top \mathbf{x}_j) \right) \tag{19}$$

Quando o problema é formular aos moldes das restrições 18 e 19, tem-se uma *hard margin* (margem rígida) a qual impõe que todos os pontos de treinamento sejam corretamente classificados dentro da margem. Contudo, os dados podem conter ruídos ou serem intrinsecamente não separáveis linearmente, tornando essa abordagem impraticável.

Para lidar com essas limitações, pode-se adotar a estratégia de *soft margin* (margem suave), permitindo a violação das restrições para alguns pontos. Nela, introduzem-se variáveis de folga (ξ_i) para permitir que alguns pontos estejam dentro da margem ou até mesmo sejam classificados incorretamente. O problema de otimização é reformulado como:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i \tag{20}$$

Sujeito a:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i \tag{21}$$

$$\xi_i \geq 0 \quad \forall i \tag{22}$$

Em que C é um parâmetro que controla o trade-off entre maximizar a margem e minimizar o erro de classificação. Valores maiores de C dão mais ênfase à minimização dos erros, enquanto valores menores permitem margens maiores com mais violações. [3]

A formulação da tarefa de otimização da abordagem *soft margin* apresenta maior complexidade, contudo, por meio artifícios matemáticos este se reduz ao problema dual com diferentes restrições impostas.

Uma vez que o problema dual é resolvido, a fronteira de decisão da Máquina de Vetores de Suporte é definida como:

$$w = \sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b \tag{23}$$

Em que w é o vetor de pesos, x é um ponto de teste e b é o termo de viés.

O termo b é determinado pelos vetores de suporte, que satisfazem:

$$y_i(w^T x_i - b) = 1 \quad \Rightarrow \quad b = w^T x_i - t_i \tag{24}$$

Contudo, quando os dados não são linearmente separáveis, a Máquina de Vetores de Suporte utiliza o *kernel trick* para mapear os dados de entrada em um espaço de características de maior dimensão, onde é possível encontrar um separador linear. Isso é feito substituindo o produto escalar $\mathbf{x}_i^\top \mathbf{x}_j$ no problema dual por uma função *kernel* $\phi(\mathbf{x}_i, \mathbf{x}_j)$, que calcula o produto escalar no espaço transformado sem realizar explicitamente a transformação. As funções *kernel* mais comuns incluem:

- Linear: $\phi(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$
- Polinomial: $\phi(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + c)^d$
- Base Radial: $\phi(\mathbf{x}_i, \mathbf{x}_j) = e^{(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)}$
- Sigmoide: $\phi(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i^\top \mathbf{x}_j + \theta)$

III. Resultados

A. Resultados da Análise Exploratória e do Pré-Processamento

A assimetria da distribuição apurada dos dados reduzidos fornecidos tem amplitude de -0.01 a 24.94, com média de 7.93. Assim, transformou-se os dados utilizando a técnica Box-Cox para reduzir a assimetria da distribuição dos dados, aplicando-os a todos os preditores numéricos, somando uma unidade a todas os valores de todos os preditores numéricos para que sejam estritamente positivos. Após a transformação, a amplitude fica entre -0.26 a 11.76, com média de 5.3.

Como é possível observar com a matriz de correlação presente em [2], existem preditores que ainda apresentam alto grau de correlação. Assim, retirou-se 20 preditores, que apresentavam média de correlação absoluta maior ou igual a 0.75.

B. Resultados da Regressão Logística

O modelo de regressão logística aplicado foi avaliado conforme os seguintes parâmetros: Nº de Dobras = 10; Nº de Iterações = 1000; Taxa de Aprendizado = 0.001.

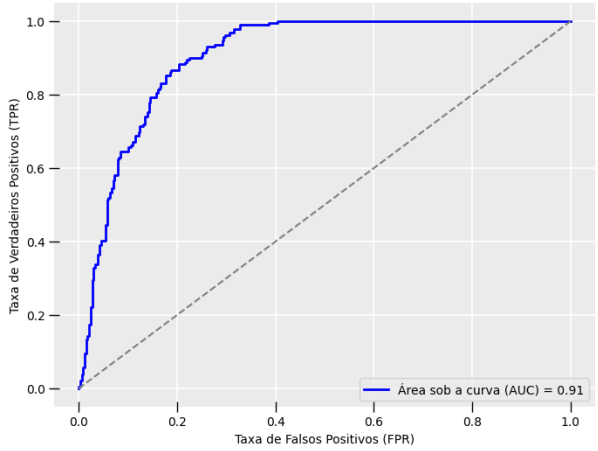
Foi alcançado uma acurácia de 0.8278 no treino e 0.8223 quando utilizada a validação cruzada com 10 dobras. Essa performance é satisfatória e sugere que o modelo foi capaz de generalizar bem os dados, equilibrando viés e variância de forma adequada.

TABLE II
MATRIZ DE CONFUSÃO PARA REGRESSÃO LOGÍSTICA

		Rótulo Verdadeiro	
		0	1
Rótulo Predito	0	259	70
	1	22	167

Além disso, analisando a matriz de confusão na Tabela II, nota-se que o modelo apresentou 259 verdadeiros negativos, 167 verdadeiros positivos, 22 falsos negativos e 70 falsos positivos. A partir dela, a sensibilidade encontrada é de cerca de 0.70, e a especificidade deu 0.92. Isso indica que, embora a regressão logística tenha um desempenho globalmente bom, ainda há um número considerável de classificações incorretas, especialmente em relação aos falsos positivos.

Fig. 2. Curva Roc da Regressão Logística



A área sob a curva ROC (AUC = 0.91) na Figura 2 indica que o modelo possui um bom poder discriminativo, conseguindo distinguir as classes de forma eficaz.

C. Resultados do KNN

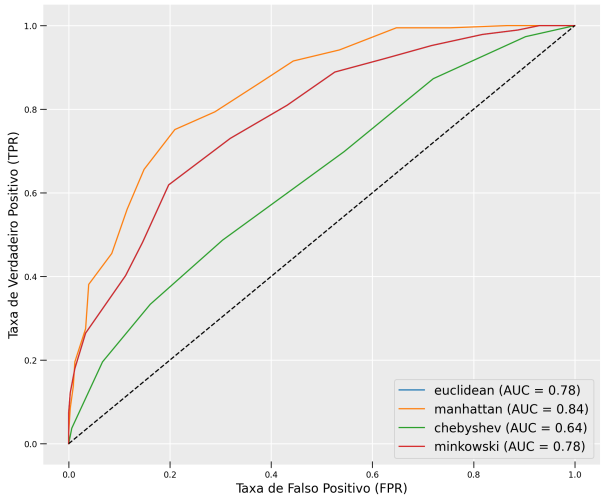
TABLE III
ACURÁCIA E MELHOR VALOR DE K

Métrica de Distância	Melhor K	Acurácia
Euclidiana	21	0.72
Manhattan	19	0.78
Chebyshev	07	0.62
Minkowski	21	0.72

Foi utilizada validação cruzada de 10-dobras para cada distância. Além disso, o parâmetro p usado em Minkowski foi igual a 2, objetivando mostrar como ele simularia a euclidiana. Como visto, o resultado para ambas foi igual.

Além disso, o uso da distância Manhattan se saiu melhor comparada a todas as outras, o que indica que a estrutura de dados favorece trajetórias ortogonais.

Fig. 3. Curvas ROC



Como esperado, a Área sob a curva ROC (AUC = 0.84), na Figura 3 que se saiu melhor foi a gerada com o uso da distância Manhattan, indicando que esse foi o melhor métrica de distância para o KNN com um bom poder discriminativo. Além disso, o gráfico de acurácia média para cada K pode ser encontrado aqui [2].

TABLE IV
MATRIZ DE CONFUSÃO MANHATTAN

		Rótulo Verdadeiro	
		0	1
Rótulo Predito	0	280	24
	1	65	124

A partir da matriz de confusão da Tabela IV temos que o *Recall* é igual a 0.84, e a especificidade foi de 0.81. Nota-se um maior equilíbrio entre esses dois parâmetros, isto indica que o modelo consegue capturar corretamente a maioria das instâncias positivas sem comprometer excessivamente a taxa de falsos positivos. Ao comparar a matriz de confusão com as geradas usando outras distâncias, foi percebido que a diferença maior se deu nos falsos positivos, indicando que o modelo conseguiu generalizar melhor e incluir mais positivos corretamente.

D. Resultados da Máquina de Vetores de Suporte

Foram utilizadas como funções para o *kernel*: linear, base radial (do *inglês radial basis function*, RBF), polinomial e sigmoide. Aplicou-se então validação cruzada de 10-dobras, obtendo-se as médias de acurácia expostas na Tabela V. Após o treinamento para cada função, os resultados obtidos com o conjunto de dados de teste estão também na Tabela V.

Com base na Tabela VI, temos uma sensibilidade de 0.81 e uma especificidade de 0.90. Dessa forma, o *kernel* linear

TABLE V
MÉDIA DA ACURÁCIA PARA DIFERENTES FUNÇÕES PARA VALIDAÇÃO CRUZADA DE 10-DOBRAS, E A ACURÁCIA COM O CONJUNTO TESTE NA SVM

Função	Média da Acurácia	Acurácia
Linear	0.8078	0.8687
Base radial	0.7869	0.8610
Polinomial	0.7056	0.7761
Sigmoide	0.7711	0.8263

TABLE VI
MATRIZ DE CONFUSÃO PARA FUNÇÃO LINEAR

		Rótulo Verdadeiro	
		0	1
Rótulo Predito	0	292	37
	1	31	158

do modelo SVM tem maior eficiência em predizer instâncias negativas.

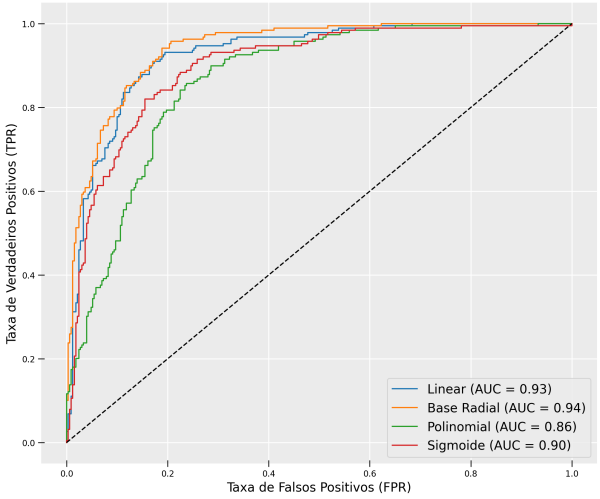
TABLE VII
MATRIZ DE CONFUSÃO PARA FUNÇÃO DE BASE RADIAL

		Rótulo Verdadeiro	
		0	1
Rótulo Predito	0	291	38
	1	34	155

Com base na Tabela VII, temos uma sensibilidade de 0.80 e uma especificidade de 0.89. Dessa forma, o *kernel* de base radial do modelo SVM tem maior eficiência em predizer instâncias negativas.

A área sob a curva ROC para a função linear foi de 0.93, para a função de base radial foi de 0.94, para a função polinomial foi de 0.86 e para a função sigmoide foi de 0.90, conforme representado na Figura 4, o que indica um bom poder discriminativo.

Fig. 4. Curva ROC para a Máquina de Vetores de Suporte utilizando *kernel* de Função de Base Radial



Observa-se que dentre as funções de *kernel* as que melhor performaram para a Máquina de Vetores de Suporte foram a linear e a função de base radial, com melhores resultados nos três parâmetros. Essa observação é reforçada pelas Tabelas VI e VII, apresentando uma ótima proporção de amostras corretamente categorizadas, porém com uma tendência a errar mais ao classificar amostras que representam aplicações verdadeiramente bem-sucedidas em relação às que são verdadeiramente malsucedidas.

E. Conclusão

TABLE VIII
RESULTADO FINAL

Métodos	Acurácia	Recall	especificidade	AUC
Regressão Logística	0.82	0.70	**0.92**	0.91
KNN (Manhattan)	0.78	**0.84**	0.81	0.84
SVM (Linear)	**0.87**	0.81	0.90	0.93
SVM (BRF)	**0.87**	0.80	0.89	**0.94**

Pode-se concluir que o conjunto de dados apresenta uma forte característica linear, visto que os resultados que apresentaram maior acurácia, especificidade foram os modelos de regressão logística e o SVM com um kernel linear. Apenas o recall do modelo KNN que se sobressaiu aos outros. Além disso, ao avaliarmos o AUC podemos ver que os de caratér linear apresentam um alto poder de discriminação entre classes, o que significa que estes são capazes de separar as classes de uma forma muito satisfatória. Possivelmente a menor performance dos modelos não-lineares quando comparado aos lineares é devido à sua flexibilidade que possivelmente os levou a se adaptar aos ruídos contidos nos dados. Em suma, o "melhor" modelo depende do contexto de aplicação, seja do que se deseja otimizar da predição - como a sensibilidade ou especificidade - ou do custo computacional.

REFERENCES

[1] W. Rodney, *Predict Grant Applications*, 2010, Kaggle [Online]. Disponível: https://kaggle.com/competitions/unimelb

[2] J. Cícero, L. David, M. Gabriel, P. Igor, *Google Colab: Códigos utilizados no Artigo.*. Disponível em: https://colab.research.google.com/drive/1Thi7FN5Awhe0KiRauyQ9oF2Q5xOvFWk_?usp=sharing

[3] Mehreen Saeed, *Method of Lagrange Multipliers*, Machine Learning Mastery, [Online]. Disponível em: https://machinelearningmastery.com/method-of-lagrange-multipliers-the-theory-behind-support-vector-machines-part-2-the-non-separable-case/

[4] IBM, *Support Vector Machine*, [Online]. Disponível em: https://www.ibm.com/think/topics/support-vector-machine

[5] Raul Fonseca Neto et al., *Um algoritmo de otimização online para SVM* , Sociedade Brasileira de Informática em Saúde, [Online]. Disponível em: https://sbic.org.br/wp-content/uploads/2016/08/CBRN2005_045.pdf

[6] Elastic, *What is KNN*, [Online]. Disponível em: https://www.elastic.co/pt/what-is/knn

[7] J. Cícero, L. David, M. Gabriel, P. Igor. *Aplicação de Modelos de Regressão Linear e Rede Neural sobre Dados de Solubilidade*. 2025. Trabalho acadêmico não publicado, UFC.

[8] M. Kuhn, K. Johnson, *Applied Predictive Modeling*, 2013rd ed., Springer, ISBN 978-1-4614-6849-3.

[9] S. Hrvoje, *Understanding Precision versus Recall: Strike the Right Balance for Effective Analysis*, 12-Abr-2024, Graphite Note [Online]. Disponível em: https://graphite-note.com/precision-versus-recall-machine-learning/

[10] L. André, *Medidas de Performance: Modelos de Classificação*, 09-Jan-2023, Brazilian AI Network [Online]. Disponível em: https://brains.dev/2023/medidas-de-performance-modelos-de-classificacao/

[11] GeeksForGeeks *AUC ROC Curve in Machine Learning*, 07-Fev-2025, [Online]. Disponível em: https://www.geeksforgeeks.org/auc-roc-curve/.