Análise de modelos preditivos para julgar a liberação de bolsas de pesquisa por pedido

1º Ramiro de Castro
Universidade Federal do Ceará (UFC)
Fortaleza, Brasil
ramirorrccastro@gmail.com

2º Glauton Santos
Universidade Federal do Ceará (UFC)
Fortaleza, Brasil
glautoncardoso@gmail.com

3º Gustavo Fechine
Universidade Federal do Ceará (UFC)
Fortaleza, Brasil
gustavofechine@gmail.com

***Resumo*—Este artigo tem como objetivo encontrar um modelo preditivo que consiga compreender a decisão sobre a aprovação de uma bolsa de pesquisa e nos permita classificar com mais facilidade futuras decisões. Na construção desses modelos, podem ser idententificados certos paramêtros que têm um maior peso no processo de decisão. Para este estudo foram usados modelos lineares e não linerares.**

I. Introdução

Em meio aos grandes avanços da internet na vida das pessoas nas últimas décadas, algumas empresas viram, nesse universo, uma grande oportunidade de obter lucro com a obtenção de dados e a classificação de usuários da rede para oferecer a outras empresas ou organizações uma chance de melhor direcionarem seus recursos de divulgação, atingindo, assim, apenas aqueles usuários que estariam mais afim dos serviços ou dos produtos destas. Além disso, muitos pesquisadores e cientistas também se utilizam de um conjunto de dados para realizar classificações dos objetos que estudam.

Porém, quando o volume de dados é muito grande, fica extremamente difícil, se não impossível, de se realizar uma classificação precisa desses dados de forma manual. Para esse trabalho foi criado o conceito de Modelos de Classificação.

Modelos de Classificação tem por principal objetivo definir um padrão de classificação para um conjunto específico de dados e, quando o sistema receber mais dados do mesmo tipo, ele pode tentar prever a qual classe esses dados pertencem. Dentre alguns exemplos de aplicação desses modelos, temos os casos onde se quer saber se determinada ação causada por um atuador irá ou não afetar em alguma outra coisa no futuro, ou de que modo irá afetar. Além disso, os modelos também podem ser utilizados quando queremos saber de forma mais fácil em qual definição se encaixa os valores de entrada ou até mesmo por pura diversão, criando inteligências artificais que desafiam humanos em jogos avançados de estratégia. As possibilidades são inúmeráveis.

Através de métodos de classificação, esses modelos podem ser gerados de forma até bem simples, o que facilita bastante o trabalho das empresas, pesquisadores ou entusiastas que atuam nessa área. Atualmente, existem vários métodos já bem definidos para realizar a criação de modelos de classificação para um conjunto de dados de treino fornecido ao sistema. Esses métodos podem ser divididos em duas categorias, os lineares e os não lineares.

Os métodos lineares são aqueles que tentam, através de uma função, criar zonas de divisão de dados. De forma simples, são aqueles métodos que, com base em um conjunto de dados de treino, tenta dividir esses dados por uma função linear, criando zonas de classificação. Esses métodos possuem toda uma matemática e cálculos envolvidos para a melhor divisão dos dados. Dentre os métodos que estão nessa categoria, este artigo abordará o método da Regressão Logística.

Também existem os métodos não lineares, que costumam ter um matemática mais complexa em seu processo. Porém, são os mais utilizados e comentados nos dias atuais. Cada um dos métodos não lineares tentam dividir os dados da sua maneira. Para este artigo, será abordado o método do *K-Nearest Neighbor*.

Para este trabalho, será abordado o desafio proposto pela Universidade de Melbourne [1], mas que é originário de um problema que afeta grande parte do mundo: a redução de fundos destinado a bolsas de pesquisa, que está diretamente relacionado a taxa de aprovação dos pedidos de bolsas feitos à universidades, que na Austrália já caíram para 20-25%. Visto isto, o desafio é criar modelos que consigam predizer caso um pedido seja aprovado ou não, para assim economizar tempo da universidade e dos alunos em relação aos pedidos com baixa chances de sucesso, além de economizar recursos e tentar encontrar fatores que tenham mais impacto nesta decisão.

II. Metodologia

Para testar a eficiência dos métodos comentados na introdução, utilizou-se um dataset que possui 8708 amostras e 1882 preditores, porém devido a alta correlação entre vários destes preditores, 1630 deles foram retirados. Dentre as amostras, 8190 delas foram separadas para o treino, dentro da variável *trainig*, e as 518 restantes foram utilizados para o teste do modelo, e estão contidos na variável *testing*.

As amostras são todas relacionadas a pedidos de bolsa feitos entre os anos de 2004 e 2008, e existe um coluna , *Class*, que contém o resultado dado pela universidade para este pedido, sendo 1 para os pedidos aprovados e 0 para os rejeitados.

Foi testado um modelo linear, a Regressão Logística, e um modelo não linear, o *K-Nearest Neighbor*, e após treiná-los e testá-los, seus resultados foram comparados por meio da Matriz de Confusão e a acurácia das predições

A. Regressão Logística

$$\mathbf{f}(\chi) = \chi\omega + \beta_0 \tag{1}$$

$$p(\chi) = \frac{1}{1+e^{-f(\chi)}} \tag{2}$$

$$L(\beta_0, \omega) = \prod_{i=1}^{n} p(x_i)^{y_i}(1-p(x_i))^{1-y_i} \tag{3}$$

A equação **(1)** é uma regressão linear, que é utilizada por este método para classificar uma amostra. χ é a matriz de preditores, de tamanho **m** x **n**, que contém **m** amostras e **n** preditores, ω é o vetor de coeficientes (β_1,β_2,...,β_n) dos preditores e β_0 é o coeficiente de interseção.

A equação **(2)** é a probabilidade de determinada amostra ser aprovada, e esta é a função logística usada para gerar a Regressão Logística, que tem o comportamento de uma sigmoide.

A equação **(3)**[2] é a função de verossimilhança, **P**(x_i) é a probabilidade do pedido de bolsa ser aprovado, e y_i é a classificação da amostra, 1 para aprovado e 0 para rejeitada. Nesta equação procura-se os coeficientes, β_0,β_1,β_2,...,β_n, que gerem seu valor máximo, para assim encontrar o modelo que melhor separa as classes.

Os resultados obtidos utilizando os métodos citados estão na Matriz de Confusão da **Tabela 1**, e foi obtido uma taxa de acerto geral de **83,78%** , agora olhando as duas classes preditas pelo modelo, vê-se que ele acertou 86,93% de suas predições de rejeição, enquanto as aprovações tiveram um acerto de 78,30%. Isto indica que o modelo está mais propenso a classificar correntamente a rejeição de um pedido.

B. K-Nearest Neighbor(KNN)

Nessa segunda parte foi realizado um aprendizado de um modelo de classificação não linear utilizando os preditores do conjunto de treino apresentados neste relatório. Foi escolhido o método k-nearest neighbors para a realização da atividade, sua ideia central se baseia numa medidade de similaridade, normalmente uma medidade de distância, como a distância Euclidiana, para dizer a qual classe o dado pertence.

O KNN pode ser utilizado em diversos tipos de aplicações, tais como saúde, finanças, ciência política e reconhecimento de imagem e de vídeo. Para ilustrar essa situação, podemos pegar um exemplo da área de saúde, em que, dado um conjunto de dados de pacientes, o algoritmo pode classificá-lo como doente ou não doente.

O algoritmo pode ser usado tanto para problemas preditivos de classificação, em que a saída é discreta, quanto para problemas preditivos de regressão, com saída contínua. Mas seu uso é mais comum em problemas de classificação.

O funcionamento do algoritmo condiz com o nome. Tem-se um número de vizinhos mais próximos K, sendo esse o principal valor decisivo. No caso de K = 1, é declarada uma classificação especial, denominada de algoritmo do vizinho mais próximo, sendo este o caso mais simples possível, mas o K pode ser alterado a fim de se conseguir uma melhor predição, como foi feito neste relatório. Na **Figura 2** é possível observar o caso do vizinho mais próximo com clareza. Pegue o triângulo como sendo a variável G, que representa o que se quer classificar. A princípio, é necessário encontrar o ponto mais próximo de G, que seria o quadrado, a partir disso, é possível classificar G como do tipo quadrado, pois ele, em teoria, possui a classificação igual a classificação do seu vizinho mais próximo. Já se K $>$ 1, então a classe a ser escolhida é pela maioria dos vizinhos mais próximos, como pode ser visto na **Figura 3**. No caso de empate na classificação do número de vizinhos, pode ser feito um sorteio aleatório para definir a classificação.

Para realizar a tarefa, foi verificado a variância de algumas variáveis preditoras para sabe se haveria necessidade de colocar os dados numa mesma escala, para se obter um algoritmo mais performático. Foi encontrado no conjunto de treino os valores da tabela de número IV para algumas variáveis preditivas. Foi realizado o mesmo processo para o conjunto de teste, que pode ser encontrado na tabela de número V.

A partir da análise dos dados encontrados, é possível perceber que a escala não está padronizada, para isso, foi preciso realizar um procedimento de padronização da variância das classes dos conjuntos de teste e de treino. Então, após aplicar uma função de scale em cima tanto do conjunto de treino quanto do conjunto de teste, fez-se necessário verificar a variância em cima das variáveis preditivas dos conjuntos supracitados novamente, agora transformados. Logo, após esse processo, constatou-se que as variâncias estavam padronizadas com valores iguais a 1.

A seguir fez-se necessário analisar o melhor valor de K para o nosso exemplo, para isso, foi realizado um loop que verificava o percentual de erro para cada valor de K entre 1 e 100. O gráfico com o resultado pode ser visto na **Figura 1**. Ao fim deste procedimento, uma variável contendo todos os erros para valores de K entre 1 e 100 foi gerada. A análise desse gráfico nos retornou a informação de que o melhor valor de K foi 26, pois nesse ponto, o percentual de erro gerado foi o menor dentre os analisados, igual a 27,9%, o que rendeu uma taxa de acerto geral de **72,1%,** o que pode ser observado na Matriz de Confusão da **Tabela 2**. Olhando agora as duas classes preditoras, observa-se uma taxa de acerto de 81,76% nas predições de rejeição e uma taxa de 55,02% nas predições de aprovação.

III. Resultados

Tendo feito e testado ambos os modelos, ficou evidente que o modelo linear teve um desempenho melhor do que o não linear, isto pode ser justificado por não ter sido encontrado o valor ótimo para K, no método KNN. É interessante notar que ambos os modelos obteram um melhor desempenho prevendo os pedidos rejeitados, e isto é muito importante pois já se pode ter um alto grau de certeza se um pedido será rejeitado, e também pode-se usar isto para entender melhor os fatores que levam a este resultado.

Todo o código utilizados neste artigo foi feito na linguarem **R**[3], com base no livro de Max Kuhn[4], e está disponível, junto com as imagens, no repositório do github[5].

Referências

[1] Predict Grant Applications, https://www.kaggle.com/c/unimelb

[2] *TI0077 – L12_classif_logistic*, slide da disciplina de ICA

[3] Kuhn, M., Johnson, K. (2013). Applied Predictive Modeling New York: Springer.

[4] R. Development Core Team, R: a language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN: 3-900051-00-3, 2008, http://www.R-project.org.

[5] Repositório aberto no github com o código utilizado neste relatório: https://github.com/Glautor/homeworks-ica.

IV. Imagens e Tabelas de Referência

Tabela I
Peformance da Regressão Logística

Predição	Aprovado	Rejeitado
Aprovado	148	43
Rejeitado	41	286

Tabela II
Performance do KNN

Predição	Aprovado	Rejeitado
Aprovado	104	60
Rejeitado	85	269

Tabela III
Variância no Cojunto de Treino

Posição da Variável Preditiva	Variância
1	1.016723
2	0.03638266
3	0.8112048

Tabela IV
Variância no Cojunto de Teste

Posição da Variável Preditiva	Variância
1	1.271028
2	0.02634743
3	0.8651225

Figura 1. Porcentagem de erro para cada valor de K do K-Nearest Neighbor

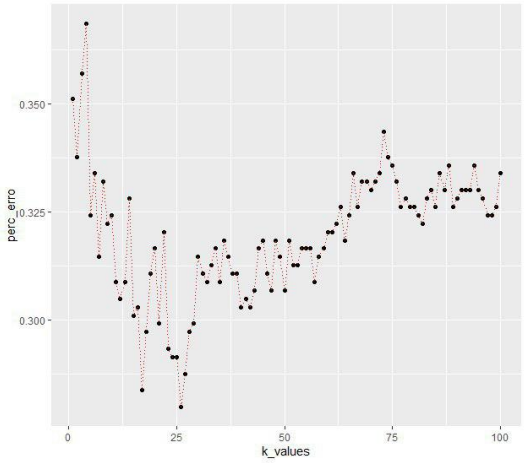

Figura 2. Algoritmo do Vizinho Mais Próximo

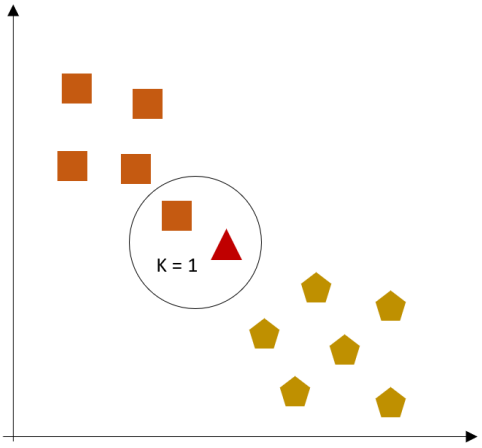

Figura 3. Algoritmo dos Vizinhos Mais Próximos