Modelos preditivos de classificação para prever o sucesso de pedidos de subsídio para a Universidade de Melbourne

1st Mario Victor Rodrigues Sales
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
mario.rs1996@gmail.com

2nd Emerson Marques Araujo
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
emerson.arj@gmail.com

3rdJoão Marcelo Pinto Ribeiro
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
joaomarcelo1812@gmail.com

***Resumo*—Este trabalho consiste em uma análise de dados de pedidos de subsídios para uma universidade(pedidos de bolsa), e incluem dados como várias características de cada indivíduo na bolsa, como seu ano de nascença, área de pesquisa, classe socioeconômica, e também outros dados, como o sucesso ou fracasso anterior em receber uma bolsa do CI(chefe investigador), o valor da bolsa, a data de submissão da bolsa, entre outros. O objetivo desse trabalho é criar modelos preditivos usando diferentes métodos de classificação para tentar prever se um pedido de subsídio seria aceito ou não pela universidade, utilizando os dados dos pedidos como preditores, e assim prover uma comparação entre os métodos utilizados.**

***Index Terms*—Ciência de dados, Inteligência Computacional, Análise Estatística**

I. Introdução

A modelagem preditiva de classificação, consiste em encontrar uma função de mapeamento de variáveis de entrada para variáveis de saída discretas(diferente da regressão que busca fazer este mapeamento para saídas contínuas). De forma geral, as técnicas de classificação procuram categorizar as amostras em grupos(também podem ser chamados de classes, ou categorias) com base nas variáveis preditivas [1].

O caminho para realizar este processo de classificação, é diferente para cada técnica. Nos métodos lineares, a classificação é realizada com base no valor de uma combinação linear dos preditores, o que exige um caminho matemático como na LDA(Linear Discriminant Analysis). E os não lineares, que são usados para separar instâncias que não são linearmente separáveis, os quais, em geral, tem uma abordagem algorítmica, como o método dos K-vizinhos mais próximos e nas Redes Neurais Artificiais [1].

O LDA(Linear Discriminant Analysis) ou ”Análise de Discriminantes Lineares”é uma técnica de redução de dimensionalidade que pode ser usada para classificação de padrões, ela procura remover os recursos redundantes e dependentes, transformando os recursos do espaço dimensional superior para um espaço com dimensões inferiores. Para isso, primeiramente, ela calcula a variância entre diferentes classes, então calcula a variância dentro das classes, e então o método consiste em construir o espaço dimensional inferior que maximiza a variância entre as classes e minimiza a variância dentro das classes. Ela é usada, por exemplo, em biometria, bioinformática e química.

As Redes Neurais Artificiais são um tipo de algoritmo de aprendizagem de máquina modelados a partir do cérebro humano, que mostram uma relação complexa entre as entradas e saídas para descobrir um padrão. Assim como os neurônios no sistema nervoso humano, os neurônios das RNA's são capazes de aprender com os dados anteriores e por meio disso fornecer respostas na forma de previsões ou classificações. Atualmente, as RNA's são utilizadas de várias formas, como reconhecimento de imagem, reconhecimento de fala, tradução automática, e até para diagnósticos médicos [1].

II. Metodologia

Os dados analisados neste artigo, obtidos no site Kaggle, consistem em 8708 observações relacionadas a pedidos de subsídio feitos entre nos anos de 2005 e 2008, dentre esses dados temos informações como várias características de cada indivíduo na bolsa, o sucesso ou fracasso anterior em receber uma bolsa do CI(chefe investigador), o valor da bolsa, a data de submissão da bolsa, entre outros. Filtrando-se os preditores que estão mais correlacionados, as variáveis D foram reduzidas a 252 (conjunto D reduzido = 252) que será o conjunto utilizado neste artigo, no qual, a saída será a variável discreta que indica se o pedido de subsídio foi bem ou malsucedido, e está representado na coluna ”Class”. Para construir nossos modelos de classificação as observações para os preditores e a classe são divididas entre conjuntos de treinamento((N tr = 8190) e teste(N ts = 518). Em nossas análises, iremos tentar estipular se o pedido de subsídio foi ou não aceito pela universidade, com base nas variáveis preditoras.

Para analisar os dados, por meio de aprendizagem estatística, e aplicar modelos de classificação lineares e não lineares, foi utilizada a linguagem de programação R, uma linguagem voltada à manipulação, análise e visualização de dados. Foram usadas algumas bibliotecas externas para facili-

tar o cálculo e o plot dos gráficos, todas elas estão no arquivo do código.

Neste artigo, iremos treinar e testar dois modelos de classificação, nos quais, a saída será se o pedido de subsídio obteve sucesso ou não, dentre estes modelos, um será linear, no qual utilizaremos o método de Análise de Discriminantes Lineares, e outro não linear onde utilizaremos Redes Neurais Artificias.

- Procedimento 1

Neste primeiro procedimento, foi utilizado o conjunto de dados de treino e aplicado o modelo de classificação linear, tendo em vista que o objetivo é classificar um preditor com relação aos demais. Para tal, escolhemos o método Análise de Discriminantes Lineares ou LDA, já abordado em nossa introdução.

Para ajustar nosso modelo, usamos os preditores contidos na tabela reducedSet, pois queremos comparar o desempenho do modelo linear e do não linear usando os mesmos inputs.

Para aplicar este método, utilizamos algumas funções do R, tais como a função "lda", e a biblioteca utilizada foi a "MASS", como pode ser visualizado no código anexado ao artigo.

Neste cenário, após treinar o modelo, iremos plotar a "matriz de confusão"para o conjunto de testes, que é uma tabela que permite a visualização do desempenho de um algoritmo de classificação, pois, contabiliza o número de classificações corretas e incorretas para cada uma das classes.

Tabela I
TABELA DE CONFUSÃO PARA O CONJUNTO DE TESTES

	successful (D)	unsuccessful
successful	161	50
unsuccessful	28	279

Desta forma, para calcular a precisão do modelo, com base na matriz de confusão, calculamos proporção de padrões da classe positiva corretamente classificados em relação a todos os exemplos atribuídos à classe positiva, de acordo com a seguinte fórmula:

$$Precisão(\hat{y}(x)) = \frac{TP}{TP + FP}$$

Dito isso, a precisão encontrada pelo modelo foi de aproximadamente 84,94%.

- Procedimento 2

Neste segundo procedimento, foi treinado um modelo de classificação não linear, para fazê-lo, foi escolhido o método de Redes Neurais Artificiais. A primeira parte consistiu em treinar o modelo, utilizando os dados treino e em seguida, o modelo foi testado, utilizando os dados de teste, com o intuito identificar a performance do modelo.

Na segunda parte foi utilizada a biblioteca neuralnet que possui a função neuralnet. Foram utilizados os preditores do arquivo, reduced-set, previamente disponibilizado, que são os preditores mais relevantes para treinar o modelo. O parâmetro hidden, corresponde aos neurônios escondidos colocados nos modelo, o qual, permite utilizar um vetor para assim se obter várias camadas.

Quanto ao threshold, ele funciona como um teste de parada do modelo, no qual é um fator importante para a precisão, que, por padrão, o valor vem como 0.01. O parâmetro threshold corresponde a derivada parcial do erro da função toda vez que o modelo é treinado, uma vez que, se o valor for menor que o valor especificado a execução se encerra. Já o linear.output foi utilizado para classificação e no caso foi definido como falso neuralnet(linear.output = False). Por último, o parâmetro stepmax, que representa o número máximo de passos que a rede neural irá tomar para o funcionamento do modelo, que nesse cenário foram utilizado um valor de passos considerável correspondente a $1 \cdot 10^{15}$, uma vez que se o valor do stepmax for muito pequeno o modelo não funciona, ou seja, a rede neural não consegue desempenhar o seu papel e não retorna os valores esperados.

Outras funções utilizadas foram makePSOCKcluster() e registerDoParallel(), onde makePSOCKcluster() ajuda na alocação para integração da placa de vídeo para funcionar no processo paralelo do R, e o registerDoParallel() inicia de fato o processamento paralelo.

Após o modelo treinado, um fator importante a ser observado é a matriz de confusão como relação a classe, quanto ao sucesso e insucesso, basicamente para dizer a precisão do modelo bem como identificar características dos dados. Por exemplo, para saber se o valor que foi definido como sucesso, é de fato sucesso com relação aos demais valores atribuídos, e em seguida montada a tabela que descreve o resultado da matriz de confusão.

Tabela II
TABELA DE CONFUSÃO PARA O CONJUNTO DE TESTE

	successful (D)	unsuccessful
successful	141	62
unsuccessful	48	267

Por fim, foi calculada a precisão do modelo, com base na fórmula explicada no primeiro procedimento, desta forma, foi obtida uma precisão de aproximadamente de 79%. Para tentar reduzir o erro poderiam ser usadas mais camadas escondidas e cada uma com mais neurônios, possivelmente usar mais preditores e também reduzir o threshold, mas todas essas medidas tornam o modelo mais custoso, aumentando assim seu tempo de processamento (treino). Após executado, foi criada a imagem mostrada a seguir como resultado do modelo feito por redes neurais baseado na predição da classe, para sucesso e insucesso e sua relação com os demais preditores. Essa imagem representa a rede neural em si, com os valores e as "sinapses"entre cada neurônio.

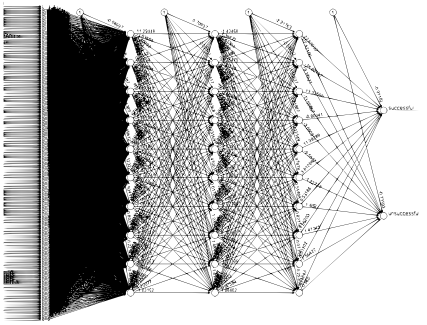

Figura 1. Plot da rede neural contendo 252 inputs (preditores), 3 camadas com 10 neurônios em cada uma, 4 constantes (uma para cada camada e uma para a ligação com as saídas) e 2 outputs.

RESULTADOS

Analisando os modelos de classificação abordados nestes procedimentos, se evidenciam algumas características de cada um, e em como é o desempenho deles em diferentes circunstâncias.

O modelo de classificação linear no qual foi utilizado o método de ”Análise de Discriminantes Lineares”, como é um método linear, tem vantagens por ser menos custoso computacionalmente, e para alguns exemplos, ele obtém um resultado aceitável, com o nosso conjunto de dados obtivemos uma precisão de 84.94%. Porém, como a sua etapa de treinamento é um processo mais matemático, não há como, por meio de mais tempo de treinamento, melhorar a precisão deste modelo.

Já quando se trata do modelo de classificação não linear, onde utilizamos Redes Neurais Artificiais, ele é bem bem mais custoso computacionalmente, por conta do tempo que leva para realizar a sua etapa de treinamento, tempo que está diretamente relacionado a precisão do modelo, quanto mais tempo, mais preciso até se chegar em um limite. Em nosso procedimento, por exemplo, com apenas alguns segundos de treinamento, ele obteve pouco mais de 60% de precisão, já com alguns minutos ele se aproximou de 70%, e com 1 hora de treinamento, sua precisão foi aproximadamente 79%.

Para os nossos dados, o modelo não linear não teve uma performance superior ao modelo linear pois enquanto a precisão do modelo utilizando o método de ”Análise de Discriminantes Lineares”foi de 84.94%, a precisão do modelo não linear utilizando uma Rede Neural Artificial foi de apenas 79%, e quanto em relação ao custo de processamento, a rede neural foi bem mais custosa, necessitando de 1 hora de treinamento para alcançar essa precisão.

REFERÊNCIAS

[1] M. Kuhn, and K.Johnson, Applied Predictive Modeling, 2013.

[2] G. James, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, 2014.

[3] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2008.