Um Estudo Comparativo Entre Modelos de Classificação para Predição da Aprovação de Recursos de Pesquisa

Artur Rodrigues Rocha Neto (431951) e Matheus Costa Mesquita Martins (371846)

Universidade Federal do Ceará - Departamento de Engenharia de Teleinformática
Campus do Pici, Acesso Público, Bloco 725, CEP: 60455-970 - Brasil

Resumo. Seja por questões econômicas, ideológicas ou políticas, o investimento em ciência vem caindo de forma relativa no mundo, em especial no Brasil. As dificuldades de se obter recursos de fomento resultam em um maior cuidado com os processos efetuados para obtê-los. Um dos artefatos mais comuns nesses processos é um documento formal de pedido de recurso, onde vários aspectos técnicos e adminstrativos relativos ao projeto/pesquisa são apresentados e colocados para aprovação por um órgão ou entidade responsável. O presente trabalho propõe o uso de modelos de classificação como ferramenta para auxiliar pesquisadores a melhor produzir tais documentos. Um conjunto de amostras de pedidos, aprovados e recusados, descritos por um gama de atributos foi usado para comparar um conjunto de classificadores, lineares e não-lineares. Uma análise em termos de precisão, sensibilidade e especificidade apontou um conjunto de modelos satisfatórios para a predição do resultado de um pedido de recurso.

1 Introdução

Segundo dados do *Global Innovation Index 2018* [1], o Brasil está na 52º posição no *ranking* mundial na categoria de investimentos em Pesquisa, Desenvolvimento e Capital Humano. $1,8\%$ do Produto Interno Bruto (PIB) é investido nessa área, valor que está abaixo da meta de crescimento nacional de $2,0\%$. O repasse para as universidades federais caiu, segundo dados mais atuais de 2017, $28,5\%$, menor patamar nos últimos 7 anos[1]. Com os repasses para pesquisa menores, faz-se necessária uma maior atenção com os pedidos oficiais de recurso para requisição de fundos junto ao Ministério da Educação (MEC).

Com esse contexto em mente, esse trabalho propõe uma ferramenta de predição capaz de detectar se um pedido de recurso seria aprovado ou não pelo setor de investimentos responsável. Foram comparados tanto modelos lineares quanto não-lineares, em função de métricas de classificação e tempo de treinamento/predição. Dada a falta de dados nacionais específicos sobre as aplicações de investimento em universidades federais, foi usado um *dataset*[2] criado em 2011 pela Universidade de Melbourne que contém 8708 amostras de pedidos descritos por 1882 atributos feitos durante o período de 2005 a 2008.

A classificação de documentos é um grande desafio dada a natureza subjetiva do objeto estudado, o que se reflete na grande quantidade de atributos associados a cada amostra. Fatores administrativos e políticos também podem influencir na decisão final. Os autores entendem que a comparação dos formatos e contextos sócio-econômicos entre as instituições australianas e brasileiras não é justa, mas acreditamos que o modelo final pode ser usado como referência para pesquisadores nacionais como ferramenta útil para melhorar a preparação dos documentos.

2 Metodologia

Nosso objetivo é encontrar um modelo preditivo capaz de classificar se um pedido de recurso para pesquisa será aceito ou rejeitado.

O conjunto de dados usado foi criado em 2011 pela Universidade de Melbourne e contém 8708 amostras de pedidos de recurso realizados durante 2005 e 2008. Cada amostra possúi 1881 descritores. Dada o volume

[1]Fonte: `https://goo.gl/2L35aa`, acessado em 24 de novembro de 2018
[2]Fonte: `https://www.kaggle.com/c/unimelb`, acessado em 22 de novemebro de 2018

relativamente grande de preditores e sua alta correlação, o *dataset* provê uma lista com 252 preditores filtrados. Foi estabelecido um *pipeline* simples de classificação contendo seleção de dados, pré-processamento, execução e avaliação. Por fim, foi realizada uma pequena investigação da capacidade descritiva da lista de preditores filtrados e um novo sub-conjunto de atributos foi proposto.

Foram testados seis classificadores: 2 lineares (Análise de Discriminates Lineares e Classificador Logístico) e 4 não-lineares (Perceptron Multicamadas, Análise de Discriminantes Quadráticos, Máquina de Vetor de Suporte [2] e k-Vizinhos Mais Próximos). A avaliação dos modelos foi feita em termos de precisão e métricas oriundas da matrix de confusão (sensibilidade e especificidade). Os experimentos foram conduzidos com a ajuda da biblioteca Python para aprendizagem de máquina *scikit-learn* [3] e da linguagem de programação estatística R [4].

O problema da classificação consiste em apontar a qual classe do conjunto de classes possível y pertence uma dada observação $\overrightarrow{x}$. O pré-processamento é um dos passos mais comuns na construção de um modelo de predição. Ele abrange técnicas de adição, remoção ou transformação do conjunto de dados [5]. Três operações de pré-processamento foram efetudas nesse trabalho: normalização simples do conjunto de dados (centralização na média e escalamento pelo desvio padrão), remoção de assimetria usando Yeo-Johnson [6] e seleção de atributos a partir da variância.

Os classificadores lineares testados foram Análise de Discriminates Lineares (LDA) e o Classificador Logístico (Logit). O LDA, ou Discriminante de Fisher, é um caso particular do classificador Análise de Discriminantes Quadráticos a partir da seguinte simplificação: assume-se que todas as observações de todas as classes possuem a mesma covariância. O Logit como classificador é também uma forma de aproximação: naturalmente um método para regressão, pode ser usado como classificador binário atribuindo-se um limiar de probabilidade que divide a predição entre as duas classes.

Os classificadores não-lineares escolhidos foram Perceptron Multicamadas, Análise de Discriminantes Quadráticos, Máquina de Vetor de Suporte e k-Vizinhos Mais Próximos. O Perceptron Multicamadas (MLP) é uma associação de vários perceptrons simples conectados entre si por sinais balizados, onde o processo de aprendizado é feito através de retropropagação ou similares. A Análise de Discriminates Quadráticos (QDA) é um modelo onde assume-se que as observações são normalmente distribuídas e que, para classificar uma amostra a uma classe, calcula-se o grau de pertencimento da mesma à distribuição associada. A Máquina de Vetor de Suporte [2] (SVM) é um mapeamento das observações em um dado espaço de forma que as classes são separadas por hiperplanos, estes calculados por diferentes funções em função do arranjo das amostras (truque de *kernel*). Por fim, o k-Vizinhos mais Próximos (kNN) é um classificador baseado em métricas de distância para definir o pertencimento de uma observação a uma classe.

A maioria dos classificadores escolhidos possuem valores de configuração dos mais distintos. Esses valores são conhecidos como hiper-parâmetros. Exemplos de hiper-parâmetros são: o fator de separação C do SVM, o número de vizinho k do kNN e o número de camadas ocultas h do MLP. A escolha desses valores na montagem de modelos preditivos representa um grande desafio [7]. Os resultados apresentados a seguir representam os classificadores calibrados com os melhores valores possíveis dos seus hiper-parâmetros. A escolha foi feita com a ajuda do algoritmo *Grid Search* implementado na biblitoteca *scikit-learn* [3]. O *kernel* do SVM que obteve os melhores resultados foi o radial.

A avaliação dos classificadores foi feita com base em três métricas: precisão (P), sensibilidade ($SENS$) e especificidade ($SPEC$). Estes são calculados a partir de valores extraídos da Matrix de Confusão:

- **VP**: Verdadeiros Positivos, pedidos aprovados classificados como aprovados
- **VN**: Verdadeiros Negativos, pedidos reprovados classificados como reprovados
- **FP**: Falsos Positivos, pedidos reprovados classificados como aprovados
- **FN**: Falsos Negativos, pedidos aprovados classifciados como reprovados

Precisão é a taxa de acerto geral. Sensibilidade mede o quanto as observações Verdadeira Positivas foram corretamente classificadas como tal. Especificidade é o análogo da sensibilidade, mas em relação aos Verdadeiros Negativos (formulações em 1).

$$\begin{aligned} P &= \frac{VP}{VP + FP} \\ SENS &= \frac{VP}{VP + FN} \\ SPEC &= \frac{VN}{VN + FP} \end{aligned} \tag{1}$$

3 Resultados

Devido à grande correlação dos preditores, o conjunto de dados traz uma lista de preditores filtrados, reduzindo a quantidade de atributos de 1882 para 252. Além disso, duas formas de reduzir a dimensionalidade do conjunto de dados foram exploradas: Análise de Componentes Principais (PCA) e seleção usando variância. Uma primeira exploração nos dados foi realizada usando R e, em seguida, um estudo completo de todos os classificadores foi implementado em Python.

#PREDITORES	CUSTO	PRECISÃO (%)
1882	0,8	81,27
252	1	85,33
66 (var)	2	86,68
66 (PCA)	2	67,18

Tabela 1: Resultado preliminar SVM (*kernel* radial)

#PREDITORES	FAMÍLIA	PRECISÃO (%)
252	Binomial	83,98
252	Gaussiana	84,94
66 (var)	Binomial	83,98
66 (var)	Gaussiana	84,75
66 (PCA)	Binomial	65,83
66 (PCA)	Gaussiana	65,25

Tabela 2: Resultado preliminar Logit

Tabela 3: Resultados de investigação preliminar dos preditores

Um conjunto de atributos reduzido foi montado usando PCA em cima dos 252 atributos disponibilizados pela Universidade de Melbourne. Foram mantidas as 66 primeiras componentes, pois essas contemplavam 95% de coesão com o conjunto completo. Um segundo conjunto, dessa vez montado a partir da variância dos preditores, no R essa redução foi realizada utilizando a função "nearZeroVar", resultando em 66 preditores. Enquanto no Python foi selecionado a partir de um valor de tolerância: preditores com variância menor que 0.06 foram descartados. Esse conjunto ficou com 88 atributos. Um classificador não-linear e um linear foram escolhidos para uma exploração inicial em R desses dois novos conjuntos de dados.

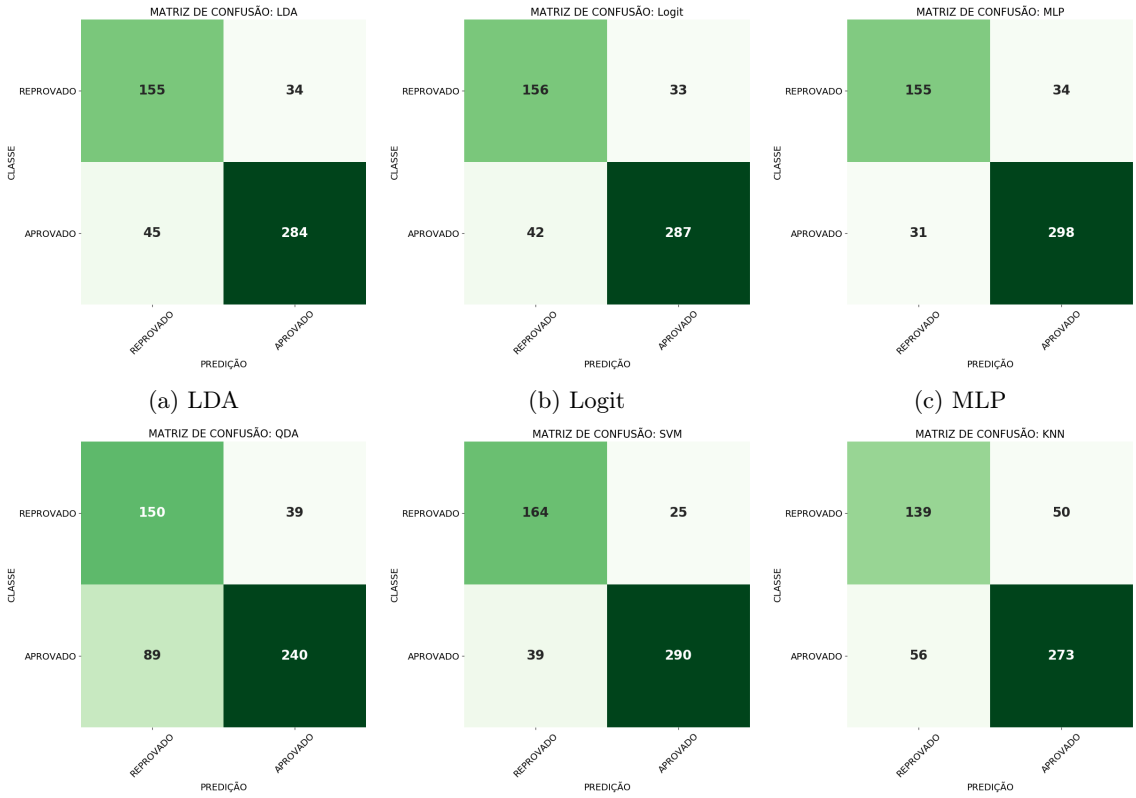
O primeiro classificador testado foi o SVM, conhecido pela sua robustez quando utilizado em conjuntos de dados de grande dimensão, ou seja, ideal para o cenário desse trabalho. Os resultados do SVM serviram para estabelecer uma primeira referência, pois são comparavéis e por vezes superiores a técnicas de apredizagem por redes neurais [8]. Foram testadas duas configurações de *kernel*: radial e sigmoidal. A função custo é ajustada até o crescimento da precisão estacionar. Com a aplicação de normalização e redução dos preditores houve melhora no tempo de processamento e crescimento na taxa de acerto. Os resultados detalhados do SVM são apresentados na Tabela 1. O segundo classificador testado foi o Logit. A Tabela 2 apresenta uma comparação entre algumas configurações montadas.

Os resultados preliminares apontam que os atributos selecionados por variância guardam maior descritibilidade que aqueles gerados por redução com PCA, tanto para SVM quanto para Logit. O SVM se mostrou

melhor que o Logit, mas por pequena margem. Uma conclusão preliminar é que a redução dos atributos usando um limiar de variância é tão bom (ou melhor) que o conjunto de atributos filtrados do *dataset*.

CLASSIF	P (%)	SENS (%)	SPEC (%)	TEMPO (s)
LDA	84,75	86,32	82,01	0,24
Logit	85,52	87,23	82,54	0,57
MLP	86,87	89,67	82,01	133,64
QDA	75,29	72,95	79,37	0,06
SVM	87,64	88,15	86,77	5,49
KNN	79,54	82,98	73,54	1,56

Tabela 4: Resultados dos Classificadores usando Python, conjunto de atributos selecionados por variância



(a) LDA (b) Logit (c) MLP

(d) QDA (e) SVM (f) KNN

Figura 1: Matrizes de confusão dos classificadores testados

A Tabela 4 traz os resultados finais de precisão, sensibilidade, especificidade e tempo total de treinamento e teste de todos os classificadores testados. Esses valores representam o uso do conjunto de atributos reduzido a partir do corte pela variância. O resultado do MLP representa uma média ao final de 100 interações. A Figura 1 mostra as matrizes de confusão geradas por cada classificador.

4 Conclusões

No geral (Tabele 4), podemos apontar o SVM como o melhor classificador em termos de Precisão, mesmo que por uma margem quase insignificante. O SVM mostrou a melhor capacidade de detectar reprovações de documentos (Especificidade de 86,77%), enquanto que a MLP foi o melhor modelo para predição de aprovações (Sensibilidadede 89,67%).

A Figura 1d mostra uma forte tendência do QDA em confudir documentos aprovados com reprovados. Esse classificador pode, portanto, apontar pedidos bem formulados como futuras reprovações. Essa confusão é mais equilibrada entre os demais classificadores. Aliando a menor taxa de Precisão, o QDA revelou o pior classificador para a tarefa.

O melhor classificador linear obtido foi o Logit. Mesmo abaixo em termos de taxa, os tempos de treinamento de teste foram mais rápidos em comparação aos demais classificadores não-lineares. Se for adicionada uma restrição de tempo à tarefa de classificação, o Logit desponta como o classificador com melhor balanço.

A montagem dos classificadores foi efetuada com bibliotecas gratúitas e abertas. Todos os experimentos foram conduzidos em computadores de configuração modesta e, ainda assim, alguns se mostraram bastante eficazes para o problema de classificação de pedidos de recurso. Os autores sugerem o uso conjunto do SVM e da MLP para a tarefa de classificação.

É importante lembrar que os vários fatores subjetivos no processo burocrático podem se elevar aos técnicos, mas os modelos apresentados representam uma boa ferramenta de auxílio e referência na elaboração de documentos de recurso.

Referências

[1] Soumitra Dutta, Rafael Escalona Reynoso, Antanina Garanasvili, Kritika Saxena, Bruno Lanvin, Sacha Wunsch-Vincent, Lorena Rivera León, and Francesca Guadagno. The global innovation index 2018: Energizing the world with innovation. *GLOBAL INNOVATION INDEX 2018*, page 1, 2018.

[2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.

[5] Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[6] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

[7] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *CoRR*, abs/1502.02127, 2015.

[8] André Carvalho, K FACELI, AC LORENA, and J GAMA. Inteligência artificial–uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, 2011.