Jomaro Correa Rodrigues

Um estudo da previsão de evasão de alunos baseado em técnicas de Mineração de Dados Educacionais aplicado ao curso de Ciência da Computação da UFPR

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Luis Carlos Erpen de Bona.

Curitiba PR
2016

Resumo

Este trabalho é uma aplicação dos conceitos de aprendizado de máquinas ao curso de Ciência da Computação da Universidade Federal do Paraná focando na previsão de evasão de alunos ou mais especificamente no problema da classificação de alunos em um grupo de risco de evasão dado o contexto histórico. O cenário parece complicado para esse tipo de experimento pois a quantidade de dados que já é normalmente pequena nos sistemas de educação tradicionais aqui é ainda mais limitada por outros fatores do curso. Neste trabalho foram feitos três experimentos, o primeiro tenta prever a evasão de alunos usando somente as notas dele ao cursar as matérias do primeiro semestre, conseguindo resultados de aproximadamente 70% tanto em F_1 *score* quanto em recall. Como a atual matriz curricular do curso é relativamente nova e portanto tem poucos dados o segundo experimento tenta mitigar esse problema usando dados do currículo anterior, o que pode na realidade não ser válido. Houve uma melhora no desempenho geral dos classificadores com muitos ultrapassando os 80% no F_1 *score* mas os melhores não tiveram grande melhora no recall. Ou seja, os classificadores devem indicar um número menor de alunos como em risco de evasão – menos falsos positivos – mas não conseguem prever a evasão dos que ele ja não conseguia antes, então de um modo geral essa abordagem se provou positiva. O terceiro experimento é uma aplicação dos mesmos conceitos porém considerando uma coleta mais tardia dos dados, quando os alunos completaram um ano inteiro no curso. Essa talvez não seja a abordagem padrão ou mais comum já que o acompanhamento ou identificação dos problemas de evasão é desejável o mais cedo possível mas ainda é uma questão válida e com suas problemáticas particulares. Os únicos resultados razoáveis foram das duas variações do NaiveBayes usadas e o QDA com resultados aproximados de 61%, 70% e 72% no F_1 score e 63%, 68% e 53% para recall.

Palavras-chave: Aprendizado de Máquina; Mineração de Dados Educacionais; Acompanhamento Estudantil.

Abstract

This work is an aplicattion of the concepts of machine learning to the course of Computer Science on Universidade Federal do Paraná focusing on the dropout of students or more precisely on the problem of classification of students in a group of risk of dropout given his historical context. The scenario looks complicated for this type of experiment because the number of data which is already small on the systems of traditional education, here is even more limited by another factors of the course. In this work were done , the first one try to predict the dropout of the students using only his grades on the first semester of the course, getting approximately 70% on the F_1 score and in recall. As the course curriculum is relatively new and gives us few data the second experiment try to reduce this problem including from the anterior curriculum, which can even not be a valid thing to do. There has been a global improvement on the classifiers with many surpassing 80% on F_1 score, but the best ones doesn't had too much increase in the recall. Which means, the classifiers should indicate a smaller number of students has in the risk group – less false positives – but can not predict the dropout of the students that it didn't could before, so this technique was proven itself positive. The third experiment is an aplication of the same concepts but considering a lately collect of the data, after the students have completed one year on the course. This may not be main approach or the most commom one since student counseling and the identification of the problems regarding the dropout is desirable as soon as it can be done, but that is still a valid question and has its own problems. The only reasonables results was the ones from the two variations of NaiveBayes utilized and the QDA with aproximated results of 61%, 70% e 72% of F_1 score and 63%, 68% e 53% on recall.

Keywords: Machine Learning; Educational Data Mining; Student Counseling.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

DINF	Departamento de Informática
PPGINF	Programa de Pós-Graduação em Informática
UFPR	Universidade Federal do Paraná
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
SVM	Support Vector Machine
SVC	Support Vector Classifier
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives

Capítulo 1

Introdução

Mineração de dados educacionais é uma área de pesquisa voltada a aplicar os conceitos e técnicas de mineração de dados (datamining) e aprendizado de máquinas (machine learning) a ambientes educacionais. O objetivo geral das técnicas desses dois campos é deixar que o computador descubra, a partir dos dados que ele processa, informações úteis aos humanos, bem como padrões que possibilitem inferências futuras quando surgirem outros exemplos seguindo os mesmos padrões.

Uma das mais interessantes e estudadas dessas aplicações é a predição da evasão escolar, reconhecida como um dos maiores problemas da educação superior brasileira e mundial. A predição da evasão, ou a descoberta de padrões que possam levar a algum conhecimento das suas causas, poderiam trazer a coordenadores de curso, professores e administradores institucionais novas perspectivas sobre seus cursos e alunos e quais mudanças poderiam ser feitas para melhorar o desempenho dos alunos.

Este é um dos objetivos deste trabalho, aplicar essas técnicas ao curso de Ciência da Computação da UFPR avaliando a problemática das mudanças curriculares como modificadoras do ambiente do curso e talvez limitadora das possíveis análises. Outros experimentos realizados tentam expandir os mesmos conceitos para a previsão da evasão após o primeiro ano de graduação.

Foi delimitado como objeto de estudo deste trabalho uma representação formada somente pelas notas dos alunos e o uso de técnicas de classificação, essas são técnicas de aprendizado supervisionado, ou seja, o modelo passa por um treinamento de "observação" de exemplos que estão associados a um marcador que caracteriza o grupo de elementos ao qual o exemplo pertence. Em um segundo passo o modelo recebe um conjunto de exemplos novos, para os quais deve tentar predizer o marcador associado. A representação escolhida parece mais detalhada em relação as usadas nos trabalhos relacionados citados pois inclui inteiramente a nota dos alunos, ao invés de uma simples média, mas sofre com problemas por depender de uma estabilidade da matriz curricular do curso.

Mineração de dados educacionais é uma área considerada nova dentro de mineração de dados que por sua vez também é considerada uma área nova, esses fatores são bastante evidentes

quando comparamos a área de mineração de dados educacionais com outras áreas similares como *Learning Analytics* que tem objetivos, técnicas e métodos bastante similares

Este trabalho fazia parte de um esforço maior na construção de um sistema online de relatórios e análises de dados construido em Python, usando a biblioteca Django como ferramenta base para a plataforma web e a biblioteca pandas para processamento dos dados, substituindo inclusive o uso de sistemas de banco de dados. O projeto foi abandonado por algumas questões técnicas como dificuldades com o novo paradigma de programação imposto pela biblioteca pandas e dificuldade de implementação de alguns mecânismos que tornariam a geração de relatórios bastante personalizável, e outros problemas quanto à manutenção do sistema, pois seria difícil treinar novos integrantes em todos as bibliotecas e técnicas utilizadas para dar suporte ao sistema e extendê-lo, o que é uma grande preocupação pois as necessidades de uns ou outros relatórios muda conforme o curso e o surgimento de novas regras da universidade ou leis. Dadas essas condições, o sistema ainda nem estava pronto e já demonstrava baixa manutenibilidade, em uma situação em que a manutenibilidade é uma característica chave para o seu sucesso. Ele foi deixado de lado em favor de uma iniciativa do PET Computação – iniciativa da qual o projeto foi inicialmente derivado – desenvolvida com métodos mais "ortodoxos" e de conhecimento geral para os programadores da área, como as ferramentas padrões da ferramenta Django e o uso de bancos de dados tradicionais. Os relatórios desenvolvidos para o projeto abandonado devem ser incluidos no projeto em andamento em suas primeiras versões.

Capítulo 2

Revisão de Literatura

Modelos de comportamento e previsão de resultados acadêmicos de estudantes estão presentes em várias áreas relacionadas ao ensino ou que estudam o comportamento humano e server como base para análises sobre os alunos e os meios pelos quais estes aprendem. O mais famoso talvez seja o modelo de Tinto, que inclui dados sobre a educação dos pais, dados médicos, sobre a integração social do aluno, atributos individuais, interações sociais, aconselhamento estudantil, entre outros.

Nem todos os dados usados por esses modelos estão disponíveis dentro da universidade e alguns são protegidos pela administração por questões éticas ou não são acessíveis simplesmente pela burocracia. Existe então um esforço de conseguir bons resultados com um subconjunto de dados mais facilmente obtíveis, preferencialmente internos ao curso e que não exijam que o aluno entre com suas informações ou informações complementares. O uso desses dados internos viabiliza então o uso recorrente de análises sobre o desempenho dos estudantes sem a necessidade do preenchimento de formulários exaustivos.

[Romero e Ventura, 2013] é uma boa leitura sobre a história e desenvolvimento da área de *Educational Data Mining* até 2013 descrevendo como as coisas evoluiram dos modelos sociológicos usando técnicas mais conhecidas na computação, como as aqui citadas *Machine Learning* e *Data Mining*.

A melhor análise encontrada falando sobre a UFPR sob uma ótica externa a análise de dados foi [CECHET, 2013], estudando como é o primeiro ano de faculdade para um grupo de estudantes de vários cursos. Os problemas apontados foram a falta de informação do que se espera de um aluno do ensino superior e falta de informações por parte da universidade, dificuldade de integração no curso e com os colegas e a dificuldade de se adaptar ao rítmo da faculdade e concilia-la com outras responsabilidades.

Bons materiais como revisões do estado da arte são [Baker e Yacef, 2009], [Baker et al., 2010] e [Romero e Ventura, 2010] que apesar de antigos ainda retratam bem o nicho estudado aqui sem deixar de citar outros campos de estudo.

O principal foco da área denominada *educational data mining* é criar e estudar técnicas sobre as quais resultados possam ser obtidos mas está longe de definir um método definitivo

que possa ser globalmente utilizado gerando o melhor resultado possível. De fato, espera-se que o ambiente educacional sofra mudanças com o tempo e que hajam diferenças entre cursos e instituições que causem variações nos resultados caso mesma técnica fosse aplicada às duas situações. Essas diferenças poderiam se dar por fatores determinados por exemplo pela economia local, por fatores mais específicos como a quantidade de alunos no curso, a própria configuração das disciplinas do curso, e também por fatores pessoais do aluno. Todos os experimentos estudados tem uma listagem de quais técnicas funcionaram melhor e os resultados apontados são variados, é comum que as melhores técnicas de classificação se alternem entre NaiveBayes e Árvore de Decisões por exemplo, reforçando que cada um interessado nessas técnicas deve aplica-las ao seu contexto e decidir individualmente sobre a aplicação delas.

Boa parte dos mesmos algorítmos de classificação usados aqui foram também aplicados por [Manhães et al., 2014b] em uma análise similar de vários cursos da UFRJ. Também descreve a arquitetura de uma ferramenta para fazer predições sobre o sucesso estudantil através de uma votação entre vários classificadores e o melhor classificador individual foi a árvore de decisão com uma pequena vantagem sobre os demais. Os resultados chegam aos 90% de acurácia mas parece ter sofrido também algum efeito de overfitting pois os resultados de *True Positives* e *True Negatives* tem uma diferença considerável. [de Brito et al., 2015] e [Sales et al., 2015] são estudos similares usando basicamente o mesmo vetor de características. O primeiro conseguindo taxas de TP e TN quase equivalentes girando em torno dos 85%, esse trabalho também citou a existência de mais de uma grade curricular porém analisou apenas os dados das disciplinas comuns aos três curriculos ignorando as outras matérias. E o segundo estudo obteve F_1 score aproximado de 75% e um número variante entre 71% e 94% de recall com uma base de dados de tamanho similar.

Um trabalho tentando predizer mais genericamente o desempenho do aluno com técnicas similares é [Kabakchieva, 2013] que monta categorias de acordo com um critério de classificação de alunos em 6 categorias usado pelo sistema educacional da Bulgária. Apesar do grande tamanho da base de dados os resultados são razoavelmente fracos com desempenhos pífios para algumas classes menos representadas no conjunto de dados e ficando próximo do 70% de precisão e taxa de TP nas outras.

Além de tentar prever o desempenho do aluno, outro objetivo de estudos é permitir o uso dessas técnicas por parte de orgãos institucionais de aconselhamento estudantil para os quais seriam muito úteis as técnicas automatizadas de classificação em grupos de risco que o campo da descoberta de conhecimento tem a oferecer. Algumas das técnicas de prendizado de máquinas usadas aqui foram também usadas por [Dekker et al., 2009] sob a supervisão de um conselheiro estudantil que avaliou os resultados obtidos e os casos de falha de classificação e propôs melhorias dos modelos usados. Também teve a oportunidade de testar combinações de dados incluindo ou não dados pré-universidade e do exame de admissão do aluno.

Dado o fluxo pequeno de dados gerado nos ambientes de educação tradicional esta área tem se agitado mais em volta de ferramentas de ensino. Essas ferramentas que são o foco de

outras áreas como sistemas de suporte ao aprendizado e sistemas tutores inteligentes servem como mediadoras do acesso do aluno ao conteúdo e aos exercícios e tem a possibilidade de gerar dados mais frequentemente e com uma granularidade muito mais fina sobre o aluno, seu engajamento na atividade e seu desenvolvimento. Essas aplicações são um tema mais central na área correlata de *Learning Analytics*. Uma comparação das duas áreas foi realizada e [Siemens e d Baker, 2012] traçando um paralelo entre as duas áreas citando seus pontos e interesses em comum e pedindo por esforços de colaboração entre as áreas.

Capítulo 3

Materiais e Métodos

Este trabalho aplica as técnicas de descoberta de conhecimento em uma base de dados com informações estritamente acadêmicas – ou dados administrativos como são as vezes citados na literatura – dos alunos do curso de Ciência da Computação da UFPR. A base possui dados de notas e de quando os alunos cursaram determinadas matérias, mas não possui nenhum tipo de informação sócio-econômica sobre eles. Para conhecimento, os dados foram extraídos do sistema SIE usado na Universidade Federal do Paraná e desenvolvido inicialmente para a Universidade Federal de Santa Maria no Rio Grande do Sul. Talvez o mesmo sistema seja usado ainda em outras instituições de ensino superior do pais.

A ferramenta utilizada nos experimentos é o scikit-learn [Pedregosa et al., 2011], biblioteca popular para aplicações de aprendizado de máquina em Python e que tem crescido bastante nos últimos anos. Um dos motivos dessa escolha de ferramentas é justamente pela simplicidade e facilidade de replicação do ambiente ainda que a natureza dos dados talvez não permita sua publicação para uma possível replicação integral dos experimentos.

3.1 Classificação

O problema atacado neste trabalho foi o de indicação de alunos em risco de evasão através de classificação. A classificação é um processo de aprendizagem supervisionada onde se quer delegar indivíduos a grupos antes de se saber de fato a que grupo o indivíduo pertence, consistindo então de uma técnica de predição. Nesse trabalho o problema é delegar ao aluno um dos grupos “evasão” ou “sem evasão”, o que é uma situação difícil sendo uma tentativa de previsão do futuro em um ambiente onde muitos fatores podem influênciar o resultado.

Um classificador funciona em duas etapas: treinamento e classificação. Na primeira etapa o classificador é alimentado com os dados já obtidos – ou algum subconjunto desejável desses – chamados de dados de aprendizado os quais se sabe a que grupos estão associados através de um identificador de grupo, chamado rótulo, – ou o inglês *label* – a seu modo o classificador irá identificar padrões e similaridades nos dados montando um conjunto de regras chamado modelo.

A segunda etapa, a classificação, é feita alimentando o classificador já com o modelo montado a novos dados, esses os quais deseja-se prever a associação dos grupos.

Se existem dados de aprendizado cobrindo todas as possibilidades de dados de entrada não é preciso tentar adivinhar, podemos memorizar os dados e os resultados esperados em uma tabela e então pesquisamos nela aqueles parâmetros quando uma situação idêntica acontecer. Mas isso em geral não é verdade, um classificador deve ter um bom poder de memorização, mas balanceado com uma boa capacidade de generalização pois deve indicar um resultado avaliando dados que em geral são apenas similares aos observados na etapa de classificação. Casos em que o classificador vira um memorizador com pouca capacidade de generalização são conhecidos como casos de *overfitting*.

3.2 Validação

Porém, como avaliar se o classificador e o modelo gerado têm bons resultados? Para se ter essa informação seria necessário comparar o resultado de algumas predições do classificador com resultados reais, então usa-se o classificador para prever alguns resultados e depois observa-se se eles se tornam realidade? Isso em geral não é possível ou é muito custoso. Uma técnica para fazer essas medições é chamada validação cruzada. Usa-se apenas uma parte dos dados na etapa de treinamento – em geral algo como 65% a 80% da quantidade de exemplos – e usa-se o modelo gerado para prever o resultado do restante dos dados, como sabemos o rótulo real associado a esses exemplos podemos compará-los. Para evitar que o resultado seja enviesado por uma coincidente divisão oportuna dos dados esse processo costuma ser repetido várias vezes com diferentes divisões do conjunto de dados.

Esse processo de comparação em problemas de classificação binária produz a princípio quatro métricas, descritas contextualizadas com o problema:

TP - True Positives alunos que sairam do curso e foram corretamente classificados;

FP - False Positives alunos que não sairam do curso, mas foram classificados como evasores;

TN - True Negatives alunos que não sairam do curso e foram corretamente classificados;

FN - False Negatives alunos que sairam do curso, mas foram classificados como não evasores;

Essas métricas são em geral compiladas em uma *matriz de confusão* como exemplificado na Figura 3.1 ou usadas no calculo de outras métricas para descrição dos resultados.

A métrica a principio usada nos ensaios para este trabalho, que também é a métrica padrão do scikit-learn, é a acurácia, definida como:

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.1}$$

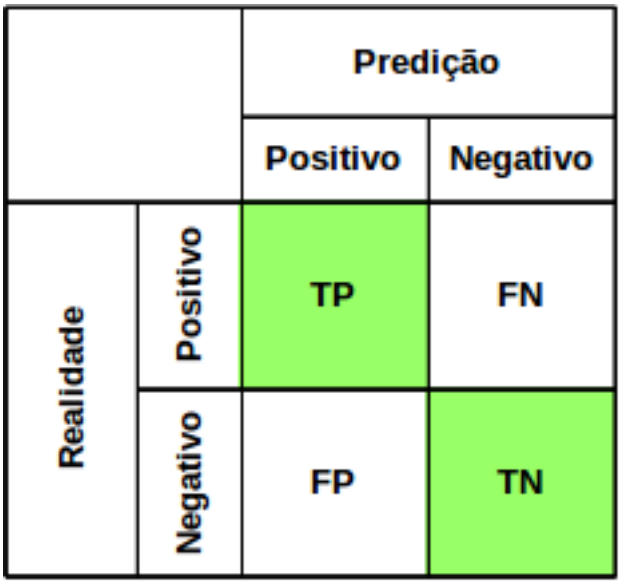


Figura 3.1: Exemplo de matriz de confusão binária

Porém o uso dessa métrica pode esconder um comportamento indesejado do classificador quando a quantidade de exemplos positivos e negativos é diferente e/ou quando temos uma classe crítica, ou seja, damos mais importância à correta classificação dos elementos de uma classe. Por exemplo se existem 100 exemplos positivos e 900 negativos e o classificador resolve simplesmente classificar todos os exemplos como negativos, teremos uma Acurácia $A = \frac{0+900}{0+0+900+100} = 90\%$ o que deveria ser considerada uma medição muito alta para um classificador que na realidade não faz nenhum esforço para classificar corretamente uma das classes, que poderia ser a classe crítica do problema.

Métricas mais relevantes são a precisão e o recall – algumas vezes traduzido como "revocação" – definidos nas equações 3.2 e 3.3 respectivamente. Segue a definição intuitiva dessas duas métricas, segundo a documentação da biblioteca scikit-learn em tradução livre:

Precisão A precisão é intuitivamente a habilidade do classificador não definir como positiva uma amostra que é negativa;

Recall O recall é intuitivamente a habilidade do classificador de encontrar todas as amostras positivas.

Essas são métricas mais específicas e é interessante que não tratam igualmente as duas classes estudadas, e especialmente comparam os acertos na classe positiva com os erros possiveis, não dando muita atenção à taxa de sucesso na outra classe. Mas nenhuma das duas é ainda uma métrica que possamos usar em um processo de otimização de classificadores ou no processo interno do classificador de construção do modelo, as duas juntas são bastante representativas, mas individualmente são enviesadas. Por isso a métrica escolhida neste trabalho é a F_1 *score*

definida na equação 3.4 como a média harmônica entre a precisão e o recall, e é também a métrica sugerida pelo scikit-learn para o caso de classes desbalanceadas.

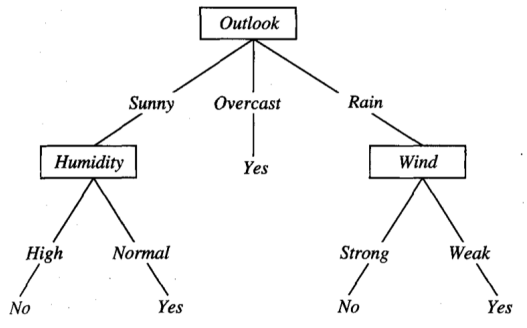
$$P = \frac{TP}{TP + FP} \tag{3.2}$$

$$R = \frac{TP}{TP + FN} \tag{3.3}$$

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{3.4}$$

3.3 Árvore de Decisão

Árvores de decisão é uma das principais famílias de algoritmos de classificação, ela consiste na montagem se uma sequência de regras de decisão IF-THEN que pode ser vista como uma árvore. Para a montagem da árvore cada uma das propriedades é percorrida e é calculado um critério – por exemplo a entropia – entre a propriedade e o rótulo das amostras, a propriedade que obtiver o melhor conceito no critério é definido como o nó atual, e o processo é repetido para cada um dos dois ramos deste nó porém usando apenas os dados que satisfazem as restrições até aquele ramo – restrições dos nós anteriores. O algoritmo deve parar após algumas iterações para evitar overfitting, um critério comum de parada é o tamanho da árvore.



(a) Forma de árvore

```
IF Outlook = Overcast THEN yes
IF Outlook = Sunny AND Humidity = High THEN No
IF Outlook = Sunny AND Humidity = Normal THEN Yes
IF Outlook = Rain AND Wind = Strong THEN No
IF Outlook = Rain AND Wind = Weak THEN Yes
```

(b) Forma de regras

Figura 3.2: Regras de decisão para o clássico problema “PlayTennis” introduzido por [Quinlan, 1986]

3.4 Naive Bayes

O Naive Bayes ataca o problema da classificação de maneira diferente, ao invés de tentar responder a pergunta “a qual classe uma amostra com essas características pertence?” ele muda a pergunta para “qual a probabilidade de aparecer um elemento com essas características nessa

classe?" e calculando para todas as classes respondemos a primeira pergunta escolhendo aquela que tem a maior probabilidade.

Isso é feito com a aplicação do teorema de Bayes:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)} \tag{3.5}$$

Como essa conta será feita para os diferentes valores de y o termo $P(x_1, \ldots, x_n)$ que não inclui y em sua formulação é constante para todos eles e pode ser desprezado, $P(y)$ é a proporção de exemplos com a label y na base de aprendizado, e $P(x_1, \ldots, x_n \mid y)$ pode ser desmembrado em $P(x_1 \mid y) \cdot P(x_2 \mid y) \cdot \ldots \cdot P(x_n \mid y)$ e como calcular cada um desses termos é o que cria as diferentes versões do classificador. As duas formas usadas neste trabalho são apresentadas abaixo.

3.4.1 Forma Gaussiana

Essa é a forma usada nos dois primeiros experimentos desse trabalho representado como GaussianNB nas tabelas de resultados por ser o nome usado no scikit-learn. Ela deve ser usada quando os dados são contínuos e o cálculo da probabilidade de um termo consiste do cálculo da média e desvio padrão dos dados de cada propriedade para uso da fórmula de probabilidade considerando uma distribuição normal.

$$P(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}}^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.6}$$

3.4.2 Forma Simbólica ou Multinomial

Essa variação usa uma forma de cálculo onde se assume que os dados sejam simbólicos ou que possam assumir um conjunto bem definido de valores e que não haja uma relação de grandeza entre as diferentes propriedades. É uma das variantes usadas em classificação de texto onde se deseja classificar um texto entre áreas de conteúdo (exemplo: química, direito, filosofia) de acordo com a quantidade de vezes que algumas determinadas palavras acontecem no texto.

Na etapa de treinamento para cada combinação de propriedade x e rótulo y o classificador monta uma tabela indicando quantas vezes um determinado valor ocorre para aquela combinação. Assim a função $P(x \mid y)$ vira pouco mais do que um consulta em tabela com a divisão do número encontrado pela quantidade de exemplos da classe.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

(a) colunas “Weather” e “Play” do problema PlayTennis

Tabela de frequências		
Weather	**No**	**Yes**
Overcast		4
Rainy	3	2
Sunny	5	9

(b) tabela de frequencias para a coluna “Weather” do problema PlayTennis

Figura 3.3: Exemplo de tabela de frequencias usada Naive Bayes Multinomial

3.5 Análise Discriminante Linear e Análise Discriminante Quadrática

A Análise Discriminante Linear – do inglês *Linear Discriminant Analysis* (LDA) – usa uma função linear para separar os dados das amostras de modo que se a equação resulta em um número maior do que zero a amostra pertence a uma classe, caso contrário pertence a outra classe, essa função pode também ser representada como um hiperplano ou um reta em uma situação com um vetor de características de tamanho 2.

$$g(x) = W^t X + w_0 \tag{3.7}$$

Onde W é o vetor dos pesos do função que determina o quanto cada uma das características influenciará no resultado final. X é o vetor de características da amostra a ser classificada e w_0 é chamado de bias e representa a distância da reta da função para a origem. O que resta é encontrar o melhor vetor W e o melhor escalar w_0, o que é feito com a formula:

$$W = C^{-1}(\mu_0 - \mu_1) \tag{3.8}$$

Onde C é a matriz de covariância dos dados e μ_0 e μ_1 são os as médias das caraterísticas agrupadas por classes tendo quantidade de posições igual ao tamanho do vetor de características. E então w_0 pode ser conseguido com algoritmos de otimização.

Mas o LDA funciona com as premissas de que os dados de todas as classes obedecem uma distribuição normal e possuem igual covariância caso no qual muitos termos da sua formulação

matemática podem ser simplificados. Caso a premissa da covariância não seja verdadeira temos o caso da Análise Discriminante Quadrática – do inglês *Quadratic Discriminant Analysis* (QDA) – que em termos simplificados é resumido na formula indicando também que a superfície de separação dos dados será algum tipo de cônica como uma linha, círculo, elipse, parábola ou hiperbole. Como ainda temos o caso linear podemos dizer que o QDA é uma generalização do LDA.

$$g(x) = x^t Ax + W^t x + w_0 \tag{3.9}$$

$$g_k(x) = -\frac{1}{2}\log|C_k| - \frac{1}{2}(x-\mu_k)C_k^{-1}(x-\mu_k) + \log p_k \tag{3.10}$$

3.6 Máquinas de Vetores de Suporte

Máquinas de Vetores de Suporte – do inglês *Support Vector Machines* (SVM) – é um classificador bastante diferente dos anteriores, enquanto os outros tem uma forma fechada de cálculo o SVM usa um componente a mais, o "kernel". Durante a classificação o SVM aplica a função kernel em uma tentativa de aumentar a quantidade de dimensões dos dados tornando-os mais propensos a serem separados, esse processo se chama "kernel trick". O SVM então indentifica os "pontos difíceis", aqueles que estão mais próximos de um grupo de pontos de outra classe, e busca um hiperplano ótimo que separe esses pontos através de um hiperplano original aleatório e algorítmos de otimização.

Para a preparação do modelo o classificador passou por um processo de otimização de hyperparâmetros, testando inclusive as várias opções de kernels diferentes, o kernel RBF foi o melhor em todos os testes. Nos experimentos além do classificador SVC foi usado também o classificador LinearSVC que é uma versão do SVC com o kernel Linear o qual não foi testado no processo de tunagem de hiperparâmetros por diferenças na API.

3.7 K vizinhos mais próximos

O "K vizinhos mais próximos" – do inglês *K-Nearest Neighbors* (KNN) – é um algoritmo de classificação por comparação, ele guarda inteiramente a base de dados de treinamento e a cada amostra X para classificação ele percorre todos os exemplos de treinamento calculando sua distância para X e seleciona os K elementos mais próximos de X, chamados de vizinhos. O rótulo definido por ele será o que mais aparece nos exemplos selecionados. O KNN pode receber então dois parâmetros: a quantidade K de elementos que devem ser selecionados, em geral para casos de classificação binária é interessante que seja um número ímpar para evitar os casos de empate na quantidade de rótulos de cada classe; e uma função para calcular a distância entre os exemplos.

Como a função usada foi uma simples distância euclidiana é necessário garantir que os dados estejam normalizados. Em um caso em que se esteja usando a distância da casa do aluno até a faculdade em quilômetros como dado para classificação ao mesmo tempo que seja usa a nota do aluno entre 0 e 1 o algorítmo vai dar uma importância desproporcional para a primeira informação. Uma alternativa de correção seria normalizar os dados previamente usando parâmetros possivelmente até arbitrários dependendo do comportamento dos dados, por exemplo usando um logarítmo da distância da moradia à faculdade para não sobrevalorizar – ou sobrepenalizar – os alunos que moram longe. Como apenas as diferentes notas do aluno foram usadas neste trabalho – já normalizadas inclusive – grandes cuidados neste sentido não foram necessários.

Como forma de otimização o KNN pode usar estruturas de dados como a BallTree e a KD-Tree para agilizar a busca dos vizinhos mais próximos.

3.8 Agregação de classificadores

Dentro dos algorítmos de classificação existem também as técnicas que consistem na junção de outras técnicas, do inglês *Ensembles* são em geral traduzidos como "agregação de classificadores" ou "combinação de classificadores". São muitas vezes chamadas de meta-classificadores pois sua definição não especifica exatamente como eles devem funcionar. Uma dessas famílias de técnicas é o *bagging* que consiste de classificadores que usam resultados médios de outros classificadores criados no processo de treinamento com subconjuntos aleatórios dos dados. A principal vantagem do uso do bagging é a diminuição da variação dos resultados que normalmente se obtém com o uso de um único classificador. Os classificadores desta família usados neste trabalho foram o BaggingClassifier e o RandomForestClassifier que é uma implementação específico para trabalhar com árvores de decisão.

O *boosting* por outro lado cria muitos pequenos classificadores que não precisam ter boa performance e os "adiciona" ao classificador final atribuindo a ele um peso de forma a melhorar a acurácia do classificador principal. Classificadores desta família usados foram o AdaBoostClassifier e GradientBoostingClassifier

Capítulo 4

Problemática

A proposta inicial deste trabalho é fazer uma primeira tentativa de aplicação dos conceitos do campo de *Educational Data Mining* no contexto da UFPR dado especificamente o curso de Ciência da Computação. A educação tradicional, que é o objeto de estudos delimitado para este trabalho, é um ambiente bastante árido para o uso de ferramentas que se propõe a obter informação a partir de dados brutos. Em uma época em que dados podem ser gerados aos milhares por minuto, os sistemas tradicionais de educação geram poucos dados por semestre, quando não anualmente. De fato, para facilidade de uso e para que as aulas ocorram sem serem engessadas por uma ferramenta o registro de informações é em geral mantido ao menor nível possível que ainda permita uma diferenciação e registro dos dados acadêmicos.

Informações sobre o progresso do aluno em geral se resumem a uma entrada no banco de dados para cada matéria que um aluno cursou naquele semestre, indicando sua nota final, frequência obtida e algum campo que indica se a matéria foi cumprida normalmente ou em alguma modalidade especial. Supondo um corpo discente de 300 alunos ativos e que cada aluno curse em média 5 matérias por semestre, a base de dados seria abastecida com 1500 registros a cada semestre, um número pequeno comparado a outros campos onde técnicas similares são utilizadas em bases de dados que recebem milhares de registros por dia.

O problema da escassez de dados pode piorar bastante dependendo de como os dados devem ser agrupados para processamento. Aqui as análises focam nos alunos, assim, cada aluno tem seus dados agrupados formando um exemplo para processamento. Novamente, supondo que a cada ano entrem 80 novos alunos no curso e temos uma base de dados sobre os últimos 10 anos, teremos apenas 800 exemplos para aprendizado cada um com apenas 5 características, uma para cada uma das matérias iniciais do curso.

Uma questão aqui abordada são as mudanças nas grades curriculares dos cursos, a princípio se uma mudança dessas acontece ela significa uma mudança considerável no ambiente estudado, e o modelo que foi criado e validado como tendo uma certa precisão ao fazer afirmações sobre o curso considerando o ambiente anterior pode não ser mais válido pois mesmo caso não se altere o o vetor de características as mudanças podem causar uma mudança no fluxo dos alunos no curso.

Um desses casos seria fazer um mudança curricular sem alterar o primeiro semestre do curso, porém colocando no segundo período as matérias consideradas as mais difíceis do curso e aplicar alguma política de penalização caso o aluno não consiga cumpri-las, essa mudança deve aumentar a dificuldade do curso e o extresse do aluno no segundo semestre e consequentemente aumentar a evasão; como enunciado essa mudança não altera o curso até o ponto em que os dados são recolhidos, mas deve alterar drasticamente a acurácia [1] do processo caso se tente aplicar o modelo antigo ignorando as mudanças.

Então se o curso passou por alguma dessas reformas curriculares recentemente e não é desejável reaproveitar dados do currículo anterior isso reduz ainda mais a coleta de dados, pois seriam necessários dados de alunos da nova grade para montar um novo modelo e supõe-se que esses dados pós-reforma sejam poucos.

Com efeito, se a reforma foi realizada a menos de 4 anos não se espera que haja nenhum aluno formado, o que é a única garantia de que o aluno não irá evadir do curso. Uma alternativa é traçar como objetivo prever se o aluno vai evadir até o quarto periodo por exemplo ao invés de tentar prever se ele vai realmente terminar o curso, isso permite o uso de dados de mais alunos e supõe-se ser válido já que evasões em períodos avançados do curso são consideradas raras. Ou como nos experimentos feitos, traçar um limiar mínimo para que os alunos sejam incluídos na construção do modelo – por exemplo, usar alunos que estejam a pelo menos três semestres na universidade – e usar a informação mais recente existente sobre sua evasão ou não do curso.

Para isso o curso escolhido para esse estudo é um bom exemplo. O curso passou por uma reestruturação curricular em 2011 depois de outra em 2007, o que faz relativamente pouco tempo mas mesmo assim é tempo o suficiente para termos um número razoável alunos em estágio avançado no curso e até mesmo alguns alunos já formados que fizeram o curso inteiro com a nova grade. Como os alunos de 2016 ainda não completaram nem 1 ano no curso estes vão ser ignorados neste estudo.

O outro complicador deste trabalho foi o desbalanceamento de classes, problema que acontece quando o número de exemplos de ums classe é consideravelmente maior que o de algum outro. Apesar da fama de ser um dos cursos de maior evasão da universidade, com taxas de evasão anunciadas as vezes acima dos 40%, dos 378 alunos válidos para análise somente 93, cerca de 25%, evadiram o curso. Menos do que o esperado e o suficiente para criar problemas com as métricas padrões dos classificadores – a acurácia – que nos testes iniciais retornavam resultados acima do esperado e acima de outros estudos realizados com bases maiores.

[1] a palavra acurácia aqui não é usada no sentido da métrica

Capítulo 5

Experimentos

Foram feitas três tentativas de classificação binária, ou seja considerando apenas as possibilidades "evasão" e "não evasão", outros trabalhos como [Manhães et al., 2012] trabalharam também com uma previsão de formatura do aluno, mas os testes iniciais desta representação trouxeram resultados considerados insuficientes. Foram utilizados vários classificadores presentes na biblioteca scikit-learn e os resultados reportam a média e desvio padrão da métrica F_1 *score* sobre 10 execuções da classificação em validação cruzada com tamanho da base de teste de 35% da base original. É mostrada também a métrica de recall calculada uma vez sobre o resultado de todas as execuções considerando "positivo" o caso de evasão.

A métrica de acurácia é uma escolha ruim para o caso de classes desbalanceadas pois não penaliza o overfitting da classe mais frequente, portanto a métrica F_1 score foi usada por oferecer uma ideia melhor do desempenho do classificador. Mas isso ainda não representa suficientemente bem o resultado da classificação, números altos ainda podem esconder um mal comportamento do classificador. O recall foi usado então como uma métrica auxiliar para a demonstração dos resultados, pois ele mostra basicamente a taxa de sucesso em classificar a classe crítica o que é visto como a parte principal do problema, "qual a porcentagem dos alunos que iriam evadir o classificador conseguiu prever corretamente?". Importante que o recall sozinho poderia esconder também um comportamento de overfitting da classe crítica, portanto ainda é necessário o uso do F_1 score ou outra métrica para auxiliar a visualização dos resultados.

Alguns classificadores passaram por um processo de tunagem de hiper-parâmetros. Hiper-parâmetros são opções usadas por alguns classificadores que podem afetar seu desempenho, alguns classificadores como o SVC praticamente dependem deste processo por em geral terem um resultado muito fraco com suas opções padrões. O processo foi feito usando o recurso GridSearchCV que testa uma a uma as combinações dos parâmetros passados.

Os experimentos foram também realizados usando os recursos do scikit-learn para dar mais importância a certos exemplos – ou certas classes – e deste modo criar modelos melhores no sentido de dar prioridade – ou ao menos não prejudicar – a classe crítica. Esperáva-se encontrar bons resultados ao definir o peso da classe crítica como 3 para 1 ou 4 para 1 invertendo a proporção de 3 para 1 que é a proporção de amostras negativas para positivas, porém mesmo

quando definida para 15 para 1 o resultado dos experimentos – visualizados em matrizes de confusão – não teve diferença mencionável, isso se deve provavelmente a falha humana no processamento dos dados.

5.1 Classificação com dados do primeiro semestre

Dadas as limitações de acesso aos dados socio-econômicos e dados escolares anteriores dos alunos, a primeira e mais imediata alternativa para predição do resultado acadêmico dos alunos é, depois dos resultados do primeiro semestre do aluno, usar suas notas como vetor de características para os algorítmos de classificação.

Em um modelo consistente, supõe-se que todas as medições que compõe o vetor de características foram feitas em um ambiente igual, assim se houve alguma mudança no curso como a troca de matérias espera-se ver uma queda no desempenho dos classificadores usados. Esse problema deve ser ainda ampliado quando essas mudanças atingem dados que fazem parte do vetor de características. No curso em estudo houve uma reformulação geral da grade curricular em 2011, com mudanças do periodo ideal de algumas disciplinas, adição e cancelamento de outras. Essas mudanças incluiram a retirada de duas matérias obrigatórias do primeiro período CI063 - MÁQUINAS PROGRAMÁVEIS e CI066 - OFICINA DE PROGRAMAÇÃO, e o deslocamento de CI068 - CIRCUITOS LÓGICOS do segundo período para o primeiro, gerando os problemas citados acima. Tabelas que representam integralmente as grades curriculares atual e antiga estão presentes nos anexos A e B respectivamente.

Nessa ideia, para coerência dos dados utilizados, nossa primeira tentativa de classificação leva em consideração apenas dados obtidos entre os anos de 2011 e 2015. Os resultados são mostrados na tabela 5.1.

Classificador	Média	Desvio Padrão	Recall
DecisionTreeClassifier	67.66%	3.77%	54.31%
GaussianNB	69.38%	7.28%	68.20%
MultinomialNB	56.84%	6.06%	34.48%
SVC	74.17%	5.81%	52.16%
LinearSVC	63.47%	17.28%	52.07%
KNeighborsClassifier	71.16%	4.41%	50.86%
RandomForestClassifier	64.32%	6.87%	48.85%
LinearDiscriminantAnalysis	70.15%	9.74%	53.46%
QuadraticDiscriminantAnalysis	72.69%	8.19%	58.99%
AdaBoostClassifier	70.28%	10.50%	59.91%
GradientBoostingClassifier	70.89%	3.83%	50.86%
BaggingClassifier	69.82%	5.84%	70.69%

Tabela 5.1: Resultados da representação com dados do primeiro semestre.

Como dito anteriormente o resultado médio do F_1 score é importante para representar o comportamento correto dos classificadores, mas não queremos apenas isso queremos que ele também – ou principalmente – resolva nosso problema, e o quão bem ele está cumprindo esta tarefa é demonstrado pelo recall, assim apenas os classificadores com um bom nível recall e F_1 score nos interessam, é importante também que o classificador tenha tido um desvio padrão pequeno nos testes, de outro modo ele pode ter um comportamento muito errático para uso prático.

Os melhores classificadores seriam então BaggingClassifier, GaussianNB e o QuadraticDiscriminantAnalysis. Os resultados são bastante satisfatórios e estão acima do inicialmente esperado para este trabalho. Alguns outros trabalhos conseguiram resultados muito superiores como os de [Manhães et al., 2014a] que chegavam a 100% de acurácia e até 100% na taxa de True Positives – o que é um outro nome para o recall – porém a maioria desses estudos tinha a seu favor bases de dados muito maiores do que a usada aqui.

5.2 Classificação usando informações da grade anterior

Dada a mudança de grade, mas uma mudança não muito grande nos elementos que compõem o vetor de características [1] uma alternativa para aumentar o número de exemplos de treinamento é usar dados da grade antiga para criação do modelo de predição. Para tanto foram pegos os dados dos alunos que entraram no curso entre 2007 e 2010, mais especificamente as notas nas quatro matérias que eram do primeiro período na grade antiga e também o são na grade nova. Porém nosso vetor de características que tinha apenas 5 elementos teria agora 4, então para complementar foram usadas as notas da matéria CI068 - CIRCUITOS LÓGICOS desde que o aluno a tivesse feito no segundo semestre – que era o período ideal desta matéria na grade antiga, – ou no terceiro semestre – com um semestre de atraso. Essa é uma adaptação um tanto estranha pois ignora um fator de amadurecimento do aluno ao considerar um aluno do segundo período do curso como se fosse um que acabou de entrar na vida universitária.

Novamente retirados os alunos em casos especiais e com dados nulos isso acrescentou 372 exemplos à base de testes. Esses exemplos não foram usados para validação pois poderiam comprometer a veracidade dos resultados. Os resultados deste experimento são mostrados na tabela 5.2.

A discussão sobre a validade ou não da técnica utilizada é deixada para o classificador no sentido de que a validação é capaz de determinar se o modelo funciona ou não, e é isso que nos interessa, saber se um classificador consegue prever os resultados corretamente. Se os resultados melhoraram não nos importa neste trabalho as questões inerentes a "porque" essa técnica funciona dadas as suas desvantagens. E houveram melhoras em alguns classificadores comparados ao experimento passado. A maior melhora foram nos classificadores anteriormente muito ruins, e que não melhoraram o suficiente para serem usados na prática. O PassiveAggressiveClassifier foi

[1] As duas versões da grade curricular estão disponíveis nos apêndices destre trabalho para comparação.

Classificador	Média	Desvio Padrão	Recall
DecisionTreeClassifier	77.50%	2.79%	53.85%
GaussianNB	79.37%	2.48%	73.30%
MultinomialNB	77.93%	4.01%	21.27%
BernoulliNB	85.23%	2.00%	47.06%
SVC	87.55%	1.66%	19.46%
LinearSVC	59.50%	30.61%	59.73%
KNeighborsClassifier	83.95%	2.78%	50.23%
RandomForestClassifier	82.28%	2.91%	49.77%
LinearDiscriminantAnalysis	84.58%	1.63%	52.94%
QuadraticDiscriminantAnalysis	85.52%	1.39%	59.28%
AdaBoostClassifier	83.92%	2.54%	47.51%
GradientBoostingClassifier	84.66%	2.38%	49.32%
BaggingClassifier	82.04%	1.72%	53.39%

Tabela 5.2: Resultados obtidos aproveitando os dados da matriz curricular anterior.

o único com resultados razoáveis, mas foi errático demais para ser considerado uma boa opção. Dois classificadores tiveram bons resultados: GaussianNB e QuadraticDiscriminantAnalysis, com grande destaque para o GaussianNB.

Em nenhum dos dois casos houve grande melhora no recall, mas houve grande melhora no F_1 score indicando que a precisão teve um grande aumento. Isso se traduz no classificador tendo a mesma taxa de acerto em selecionar alunos que vão evadir, o que é nossa questão primária, mas selecionando menos alunos que não iriam evadir – os falsos positivos –, o que resulta em um grupo menor e consequentemente menos recursos gastos em um programa de assistência a esses alunos.

5.3 Classificação usando dados do primeiro ano inteiro

Uma terceira tentativa de conseguir melhores resultados na classificação de alunos em risco de evasão é usar dados de todo o primeiro ano do aluno. Para isso serão considerados os dados de alunos que ao menos concluíram os dois primeiros períodos com a grade atual, e serão utilizados seus dados de evasão atuais – existe acesso aos dados do terceiro semestre dos alunos que ingressaram em 2015 – para treinar o modelo.

Antes, alunos que não tinham registro de alguma das matérias do período inicial foram descartados pela dificuldade de representar essa falta de informação para algoritmos que esperam um intervalo contínuo de valores e porque esses casos eram poucos. Agora porém isso já não poderia ser feito, se fossem excluidos da base todos os alunos que não cursaram alguma das matérias dos dois primeiros semestres em seu primeiro ano ficaríamos com apenas 129 exemplos na base de dados sendo 121 alunos ainda ativos e apenas 8 que evadiram após esse período.

A não realização da disciplina foi representada com o valor simbólico -1. Os dados então passam a ter características mistas de dados simbólicos e dados contínuos o que não é

diretamente suportado pelos algorítmos do scikit-learn que pode interpreta-los como contínuos, neste caso não ter feito uma matéria conta como uma grande penalidade ao aluno já que definimos para ele um caso pior do que ter cumprido a disciplina e tirado nota zero, o que pode não ser justo.

Foram considerados dois casos, no primeiro as notas dos alunos foram divididas por 5 e arrendondadas para baixo; no segundo caso o mesmo processo foi feito com uma divisão por 10. O primeiro é mais detalhado da mais informações aos classificadores, o segundo tende a gerar mais exemplos por classe o que pode reduzir a sub-representação de alguns valores na base de dados. Determinar o melhor é uma tarefa do classificador através da medição da validação.

Se o aluno cursou por duas vezes a mesma disciplina a maior das duas notas foi considerada.

Classificador	Média	Desvio Padrão	Recall
DecisionTreeClassifier	57.36%	7.45%	38.93%
GaussianNB	61.38%	7.58%	63.36%
MultinomialNB	59.75%	3.97%	71.76%
SVC	63.71%	9.27%	20.74%
LinearSVC	52.74%	4.69%	26.72%
KNeighborsClassifier	61.20%	9.24%	23.66%
RandomForestClassifier	61.71%	8.42%	25.19%
LinearDiscriminantAnalysis	64.09%	10.18%	30.53%
QuadraticDiscriminantAnalysis	68.31%	9.55%	52.67%
AdaBoostClassifier	59.54%	8.43%	26.72%
GradientBoostingClassifier	60.62%	11.29%	28.24%
BaggingClassifier	59.39%	8.69%	35.11%

Tabela 5.3: Resultados da representação usando dados do primeiro ano, com redução de 5.

Classificador	Média	Desvio Padrão	Recall
DecisionTreeClassifier	59.39%	10.62%	37.84%
GaussianNB	59.73%	4.79%	54.95%
MultinomialNB	60.38%	2.84%	68.47%
SVC	60.82%	8.02%	31.19%
LinearSVC	54.83%	6.74%	23.42%
KNeighborsClassifier	62.32%	9.32%	28.83%
RandomForestClassifier	57.50%	9.27%	29.73%
LinearDiscriminantAnalysis	68.46%	9.19%	39.64%
QuadraticDiscriminantAnalysis	65.45%	5.68%	50.45%
AdaBoostClassifier	50.90%	13.31%	37.61%
GradientBoostingClassifier	61.72%	8.86%	31.53%
BaggingClassifier	60.54%	4.25%	55.96%

Tabela 5.4: Resultados da representação usando dados do primeiro ano, com redução de 10.

Esse problema traz uma comparação interessante: qual das duas versões do Naive Bayes terá o melhor resultado, o MultinomialNB que até agora havia sido incluido apenas para completude finalmente tem um motivo para estar presente na tabela, como os dados tem características mistas entre simbólicos e contínuos e o scikit-learn não nos da nenhuma alternativa melhor para processar esses dados, ver que tipo de algorítmo tem um melhor resultado já é uma questão a ser evidênciada.

E o classificador de melhor resultado foi o MultinomialNB com resultados praticamente iguais nas duas representações, seguido pelo BaggingClassifier na segunda representação e o QDA nas duas representações. Os resultados do recall foram parecidos com os anteriores e são satisfatórios mas os classificadores pioraram quanto ao resultado geral indicado pelo F_1 score, fazendo esses modelos ainda menos consideráveis para uso prático. Esses resultados podem também trazer indagações quanto aos motivos pelos quais a divisão das amostras fica mais complicada e que outras informações deveriamos coletar do ambiente para facilitar este processo. Além é claro do problema de termos dado uma grande penalidade ao aluno caso seja tratado com algorítmos de dados sequências, o que é uma possível explicação para a grande diferença de performance entre o QDA e o LDA. O LDA que é limitado a um único hiperplano de separação entre classes não é capaz de diferenciar uma nota baixa do marcador de que o aluno não cumpriu a disciplina e é obrigado a considerá-los como uma situação só, enquanto o QDA é menos limitado e poderia tratar essa situação. Uma alternativa à marcação que a princípio deve ter mais sucesso nessa tarefa e mantém os dados em sua forma original é discutida no capítulo dedicado aos trabalhos futuros.

Capítulo 6

Trabalhos Futuros

Esse trabalho deixa vários pontos em aberto, alguns por limitação de tempo outros por algum tipo de falta de informação. A primeira sugestão a quem pretender extender este trabalho é tentar junto à administração da universidade acesso aos dados sócio-econômicos dos alunos para extensão do vetor de características. Uma tentativa de melhoria das representações usadas poderia ser normalizar a nota do aluno de acordo com as médias históricas daquela matéria com aquele professor tendo dados talvez mais relevantes.

Outra sugestão é cortar da base de dados alunos com o desempenho acadêmico muito baixo que evadiram do curso logo no começo, em especial os com grande quantidade de notas 0. Muitos alunos que desistem de uma matéria ou do curso simplesmente deixam de ir às aulas e reprovam por presença insuficiente que acarreta nota 0. Porém esses evasores podem ser facilmente identificados por um limiar aplicado às notas. Retirar esses alunos da análise permite então avaliar a capacidade dos classificadores de trabalhar além do óbvio, talvez resultando em conclusões mais interessantes.

Apesar do resultado fraco da arvore de decisão que é um classificador caixa-branca, pode ser interessante gastar algum esforço maior em obter resultados com ela e listar suas conclusões, quaisquer que sejam, na esperança de que isso forneça mais informações sobre as causas da evasão.

Capítulo 7

Conclusão

O resultado base de identificação de alunos em risco de evasão foi satisfatório e melhor do que o esperado dado o tamanho da base de dados, mas talvez não seja o suficiente para um uso prático dessas ferramentas como forma por exemplo de automatizar a indicação de alunos para um aconselhamento estudantil. Porém os classificadores "caixa-branca" que teriam capacidade de informar quais matérias têm maior relação com o desempenho futuro do estudante não atingiram desempenho o suficiente para serem usados dessa forma.

O segundo experimento que é aparentemente inédito foi bem sucedido trazendo melhoria aos resultados do experimento anterior e parece ser uma alternativa viável para cursos em situação similar. Ou seja, cursos onde houve uma mudança curricular a razoavelmente pouco tempo, mas com tempo suficiente para haver uma estabilidade após a mudança, também é importante que a mudança não tenha tornado o primeiro semestre do curso radicalmente diferente. A quantidade de matérias diminuiu sem uma mudança no conteúdo das matérias – ao menos não a ponto de isso ser divulgado – teoricamente fazendo o primeiro semestre mais leve para os alunos. Esse arranjo dos dados também desprezou qualquer tipo de fator de maturidade do aluno ao cumprir a matéria de circuitos lógicos que foi puxada para o primeiro semestre, pois supõe-se que o tempo do aluno no curso afeta seu desempenho pela seriedade em estudar – ou pelo menos, administrar seus estudos – e pela quantidade de conhecimentos relacionados que espera-se que o aluno vá acumulando durante o curso, sendo o primeiro semestre um ponto crucial.

O terceiro experimento tentou duas alternativas similares para a predição da evasão, após os alunos terem completado um ano no curso. Os resultados foram razoavelmente satisfatórios dada a pequena quantidade de exemplos, mas não o suficiente para considerar essa técnica para uma seleção autônoma séria dos alunos em risco de evasão, qualquer tentativa de aplicação dessa técnica deve passar por uma forte supervisão humana.

Referências Bibliográficas

[Baker et al., 2010] Baker, R. et al. (2010). Data mining for education. *International encyclopedia of education*, 7:112–118.

[Baker e Yacef, 2009] Baker, R. S. e Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *JEDM-Journal of Educational Data Mining*, 1(1):3–17.

[CECHET, 2013] CECHET, A. G. S. (2013). O ingresso na universidade pÚblica: AnÁlise dos sentidos atribuÍdos por um grupo de estudantes atendidos pela assistÊncia estudantil.

[de Brito et al., 2015] de Brito, D. M., Lemos, M. O., Pascoal, T. A., do Rêgo, T. G. e Araújo, J. G. G. d. O. (2015). Identificação de estudantes do primeiro semestre com risco de evasão através de técnicas de data mining.

[Dekker et al., 2009] Dekker, G., Pechenizkiy, M. e Vleeshouwers, J. (2009). Predicting students drop out: A case study. Em *Educational Data Mining 2009*.

[Kabakchieva, 2013] Kabakchieva, D. (2013). Predicting student performance by using data mining methods for classification. *Cybernetics and information technologies*, 13(1):61–72.

[Manhães et al., 2012] Manhães, L. M. B., Cruz, S., Costa, R. J. M., Zavaleta, J. e Zimbrão, G. (2012). Identificação dos fatores que influenciam a evasão em cursos de graduação através de sistemas baseados em mineração de dados: Uma abordagem quantitativa. *Anais do VIII Simpósio Brasileiro de Sistemas de Informação, São Paulo*.

[Manhães et al., 2014a] Manhães, L. M. B., da Cruz, S. M. S. e Zimbrão, G. (2014a). Evaluating performance and dropouts of undergraduates using educational data mining. Em *Proceedings of the Twenty-Ninth Symposium on Applied Computing*.

[Manhães et al., 2014b] Manhães, L. M. B., da Cruz, S. M. S. e Zimbrão, G. (2014b). Wave: an architecture for predicting dropout in undergraduate courses using edm. Em *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, páginas 243–247. ACM.

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

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of Decison Trees. *Machine Learning*, 1:81–106.

[Romero e Ventura, 2010] Romero, C. e Ventura, S. (2010). Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618.

[Romero e Ventura, 2013] Romero, C. e Ventura, S. (2013). Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27.

[Sales et al., 2015] Sales, A., Marinho, L. B. e Cajueiro, A. (2015). Predicting student dropout: A case study in brazilian higher education.

[Siemens e d Baker, 2012] Siemens, G. e d Baker, R. S. (2012). Learning analytics and educational data mining: towards communication and collaboration. Em *Proceedings of the 2nd international conference on learning analytics and knowledge*, páginas 252–254. ACM.

Apêndice A

Grade 2011

1º Periodo	
CI068	Circuitos Lógicos
CI055	Algoritmos e Estruturas de Dados I
CM046	Introdução à Álgebra
CM045	Geometria Analítica
CM201	Cálculo Diferencial e Integral I
2º Periodo	
CI210	Projetos Digitais e Microprocessadores
CI056	Algoritmos e Estruturas de Dados II
CI067	Oficina de Computação
CM005	Álgebra Linear
CM202	Cálculo Diferencial e Integral II
3º Periodo	
CI212	Organização e Arquitetura de Computadores
CI057	Algoritmos e Estruturas de Dados III
CI064	Software Básico
CI237	Matemática Discreta
CI166	Metodologia Científica
4º Periodo	
CI215	Sistemas Operacionais
CI062	Técnicas Alternativas de Programação
CE003	Estatística II
CI058	Redes de Computadores I
CI164	Introdução à Computação Científica
5º Periodo	
CI162	Engenharia de Requisitos
CI065	Algoritmos e Teoria dos Grafos
CI059	Introdução à Teoria da Computação
CI061	Redes de Computadores II
SA214	Introdução à Teoria Geral da Administração
6º Periodo	
CI163	Projeto de Software
CI165	Análise de Algoritmos
CI209	Inteligência Artificial
CI218	Sistemas de Bancos de Dados
CI220	Teoria de Sistemas
7º Periodo	
CI221	Engenharia de Software
CI211	Construção de Compilares
OPT	Optativa
OPT	Optativa
TG I	Trabalho de Graduação I
8º Periodo	
OPT	Optativa
OPT	Optativa
OPT	Optativa
OPT	Optativa
TG II	Trabalho de Graduação II

Tabela A.1: Grade versão 2011

Apêndice B

Grade 2007

1º Periodo	
CI055	ALGORITMOS E ESTRUTURAS DE DADOS I
CI063	MÁQUINAS PROGRAMÁVEIS
CI066	OFICINA DE PROGRAMAÇÃO
CM045	GEOMETRIA ANALÍTICA I
CM046	INTRODUÇÃO À ÁLGEBRA
CM201	CÁLCULO DIFERENCIAL E INTEGRAL I
2º Periodo	
CI056	ALGORITMOS E ESTRUTURAS DE DADOS II
CI067	OFICINA DE COMPUTAÇÃO
CI068	CIRCUITOS LÓGICOS
CM005	ÁLGEBRA LINEAR
CM202	CÁLCULO DIFERENCIAL E INTEGRAL II
CI202	MÉTODOS NUMÉRICOS
3º Periodo	
CI057	ALGORITMOS E ESTRUTURAS DE DADOS III
CI064	SOFTWARE BÁSICO I
CI210	PROJETOS DIGITAIS E MICROPROCESSADORES
CI237	MATEMÁTICA DISCRETA
SA214	INTRODUÇÃO À TEORIA GERAL DE ADMINISTRAÇÃO
CE003	ESTATÍSTICA II
4º Periodo	
CI059	INTRODUÇÃO À TEORIA DA COMPUTAÇÃO
CI060	SEMINÁRIOS DE INFORMÁTICA I
CI065	ALGORITMOS E TEORIA DOS GRAFOS
CI069	ADMINISTRAÇÃO DE EMPRESAS DE INFORMÁTICA
CI212	ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES
CI219	ANÁLISE E PROJETO DE SISTEMAS
CM224	PESQUISA OPERACIONAL I
5º Periodo	
CI058	REDES DE COMPUTADORES I
CI062	TECNICAS ALTERNATIVAS DE PROGRAMAÇÃO
CI211	CONSTRUÇÃO DE COMPILADORES

CI215	SISTEMAS OPERACIONAIS
CI235	ESTÁGIO SUPERVISIONADO I
SIN070	ORIENTAÇÃO BIBLIOGRÁFICA - B
OPT	DISCIPLINA OPTATIVA
6º Periodo	
CI061	REDES DE COMPUTADORES II
CI214	ESTRUTURAS DE LINGUAGENS DE PROGRAMAÇÃO
CI218	SISTEMAS DE BANCO DE DADOS
CI236	ESTÁGIO SUPERVISIONADO II
OPT	DISCIPLINA OPTATIVA
OPT	DISCIPLINA OPTATIVA
7º Periodo	
CI220	TEORIA DE SISTEMAS
CI221	ENGENHARIA DE SOFTWARE
TGI	TRABALHO DE GRADUAÇÃO I
OPT	DISCIPLINA OPTATIVA
OPT	DISCIPLINA OPTATIVA
8º Periodo	
TGII	TRABALHO DE GRADUAÇÃO II
OPT	DISCIPLINA OPTATIVA
OPT	DISCIPLINA OPTATIVA
OPT	DISCIPLINA OPTATIVA
OPT	DISCIPLINA OPTATIVA

Tabela B.1: Grade versão 2007