UNIVERSIDADE DE ÉVORA

Aprendizagem Automática - Trabalho 2

Bárbara Loureiro l48469, Ricardo Dias l48388

janeiro 2023

1 Introdução/Objetivos

O presente trabalho tem como objetivo utilizar informações do histórico académico de um conjunto de alunos para construir um modelo preditivo que possa identificar qual é o risco de abandono dos estudos. Para tal, teremos acesso a um conjunto de dados que inclui informações como curso, ECTS matriculados e concluídos e notas médias ao longo de vários semestres. utilizaremos a linguagem Python e as bibliotecas sklearn, seaborn, pickle e pandas para realizar o trabalho. As métricas de desempenho a serem consideradas serão a precisão e a cobertura, tendo como classe positiva sendo o insucesso acadêmico. Como objetivo secundário, será proposto um modelo alternativo simplificado que utilize apenas dois atributos. Este relatório inclui informação sobre as bibliotecas utilizadas, análise do conjunto de dados, descrição dos métodos utilizados e análise dos resultados obtidos.

2 Análise Exploratória dos Dados

2.1 Apresentação dos Dados

Este conjunto de dados que nos foi fornecido é composto por 2110 instâncias, representando o histórico acadêmico de alunos de diferentes licenciaturas. Cada instância possui um identificador único e informações sobre o número de ECTS matriculados, concluídos e as notas médias obtidas em diferentes semestres ao longo de 4 anos. Além disso, há também informações sobre ECTS matriculados, concluídos e notas médias obtidas há mais de 4 anos. A classe a ser prevista é o "insucesso acadêmico"(1 na coluna "Failure"do dataset). O conjunto de teste que será utilizado na apresentação terá 528 instâncias e foi extraído aleatoriamente de um dataset original com 2638 instâncias, representando 20% do conjunto total.

2.2 Análise Exploratório dos Dados

Nesta secção do trabalho iremos falar um pouco da Análise Exploratória dos Dados, as Médias de valor, distribuições dos dados, entre outros tópicos:

2.2.1 Tratamento Na's

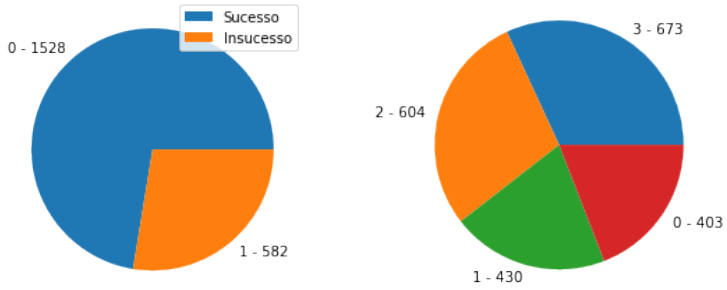
O conjunto de dados que nos foi entregue não revelou quais quaisquer Na's, portanto não tivemos que tomar qualquer decisão sobre o rumo e tratamento dos Na's.

2.2.2 Outlier's

Para procurar potenciais Outlier's utilizamos o método dos Quartis, com este método uma grande parte dos valores iriam ser removidos, então optamos por não remover outliers evidenciados por este método. Tentámos também procurar outliers analisando casos em que houvessem mais ECTS completos do que os alunos estavam matriculados mas não achamos nenhum caso do tipo.

2.2.3 Distibuição dos Dados

Os dados são desbalanceados porque A label da variável Target(Failure), que é uma variável binária, não se encontra distribuída de forma equilibrada, tendo 1528 valores 0(Sucesso) e 528 valores 1(Insucesso). Como mostra a fígura abaixo, Figura 1(a).
A variável Program, variável referente à Licenciatura frequentada pelos indivíduos, é também uma variável Categórica. Esta segue uma destruição em mais ou menos bem distribuída, existem 403 com programa 0, 430 com o programa 1, 604 no programa 2 e por fim 673 mo programa 3. Como mostra a figura 1(b) abaixo:



(a) Failures (b) Program

Figura 1: Pie Plots

Os valores das médias, desvios padrões, máximo, mínimos, etc. poderão ser encontrados nos anexos(Tabela 6). Neste podemos ver que existem variáveis com valores máximos estranhos, como por exemplo da variável Y1s1_complete que têm um máximo de 90, ou seja, esta sugere que existam alunos que num único semestre tenham completado 90 ECTS, ou seja, o equivalente a meia licenciatura convencional, 3 semestres, visto que esta métrica é uma métrica estipulada pela União Europeia para o Ensino Superior.

3 Testes

3.1 Experiência 1 -Escolha do Modelo

Num ponto inicial e com os dados já tratados e pré processados fizemos uma separação dos dados em 80% para o treino e 20% para o teste. Em seguida corremos um algoritmo que iria ver qual seria o melhor modelo para os parâmetros padrão. Este algoritmo iria treinar diversos modelos com os dados de treino/teste já mencionados. Os modelos incluem:

- DecisionTreeClassifier (Classificador de Árvore de Decisão)
- RandomForestClassifier (Classificador de Floresta Aleatória)
- BaggingClassifier
- ExtraTreesClassifier (Classificador de Árvores Extra)
- GradientBoostingClassifier
- KNeighborsClassifier (Classificador de K-Vizinhos Mais Próximos)
- GaussianNB (Naive Bayes Gaussiano)
- SVC (Classificação de Vetor de Suporte)

Estes modelos de classificação são todos diferentes, mas partilham algumas características comuns. Eles tentam todos aprender a relação entre as variáveis de entrada e a classe de saída, a partir dos dados de treino. Alguns destes modelos são baseados em árvores de decisão(DecisionTreeClassifier, GradientBoostClassifier, RandomForestClassifier e ExtraTreeClassifier), enquanto outros são baseados em algoritmos de aprendizagem supervisionada, tais como SVM. Os modelos mais sofisticados são modelos baseados em Métodos de Ensemble, como o RandomForestClassifier, o BaggingClassifier, o ExtraTreesClassifier e o GradientBoostClassifier. O algoritmo faz um ciclo por todos os modelos, ajustando-os aos dados de treino e prevendo as classes dos dados de teste. Depois disso, imprime um relatório de classificação com as medidas de precisão, recall e f1-score. Para avaliar o desempenho dos modelos, está a usar-se a validação cruzada de 5 folds (cv=5) para calcular a precisão e recall. Isso significa que os dados de treino são divididos em 5 partes (os "folds"), e o modelo é ajustado e testado cinco vezes, usando uma parte diferente como o conjunto de teste em cada iteração.

A partir das medidas de precisão e recall obtidas, é possível verificar se os objetivos do modelo estão sendo alcançados. O objetivo é maximizar o recall (taxa de verdadeiros positivos) e manter uma precisão (taxa de verdadeiros positivos sobre total de positivos) de pelo menos 70%. Isso significa que o modelo deve tentar identificar o maior número possível de instâncias positivas, sem sacrificar a precisão de forma significativa.

Através da média dos resultados obtidos pelo cross-validation, é possível comparar a performance dos diferentes modelos e escolher aquele que melhor

atende às necessidades do problema. Como esperado por nós, os modelos que tiveram melhores desempenhos para este conjunto de dados foram em geral os Métodos de Ensemble. Estes, por serem modelos que combinam vários modelos "básicos"para melhorar o desempenho do modelo final, tiveram os melhores desempenhos:

Modelo	Precisão	Cobertura
BaggingClassifier	0.84	0.71
RandomForest	0.88	0.95
ExtraTreesClassifier	0.87	0.95
GradientBoostClassifier	0.9	0.95

Tabela 1: Precisão e cobertura para diferentes métodos de ensemble com validação cruzada

Como no contexto do problema o objetivo principal seria termos uma cobertura máximo e uma precissão acima dos 70%, escolhemos o modelo GradientBoostClassifier que obteve a melhor precissão de todos e uma cobertura muito semelhante ao Modelo RandomForest e ExtraTreesClassifier. Além destes modelos de Ensemble e modelos de Classificação procurámos também otimizar os nossos modelos através de rede neuronais, nestas redes, depois de alguma otimização, obtivemos valores relativamente baixos e sentimos alguma difiduldade a treinar as nossas redes, tendo ficado estagnados. Portanto mantivemos o nosso modelo como sendo o GradientBoostClassifier. De forma geral o GradientBoostClassfieir é um modelo treinado para corrigir os erros cometidos pelo modelo anterior na sequência. O algoritmo ajusta os pesos dos modelos de forma iterativa para minimizar a perda global e maximizar o desempenho. As medidas de desempenho para os dados de teste no GradientBoostClassfier, treinando o modelo com os dados de treino 80%, foram:

Modelo	Precisão	Cobertura
GradientBoostClassifier	0.96	0.89

Tabela 2: Precisão e cobertura para diferentes métodos de ensemble para os dados de treino

3.2 Experiência 2 - Otimização do Modelo

A fim de otimizar o modelo iremos procurar qual a melhor combinação de parâmetros para conseguirmos assim maximizar a Cobertura. Para esta etapa utilizamos o método de otimização de hiperparâmetros grid search, este testa uma variedade de combinações de diferentes parâmetros e encontrar a combinação ótima que forneçam os melhores resultados. De inicio começámos por tentar otimizar todos os parâmetros mas depois de ter demorado múltiplas horas vimos que este não seria um caminho plausível e optámos por tentar reduzir o

número e a quantidade de variações dos parâmetros. Após isto verificámos que muitas das variáveis ótimas são os valores padrão e portanto apenas tentámos variar as variáveis learning_rate, max_depth e n_estimators. Após realizar o Grid Search com validação cruzada(neste caso utilizámos a validação cruzada no grid search para tentar minimizar o enviesamento nos folds) de 5 folds, obtivemos que os parâmetros ótimos seriam:

Parâmetros	**Valores**
learning_rate	0.05
max_depth	4
n_estimators	200

Com estes valores de otimização obtivemos uma precisão de 94% e uma cobertura de 89% nos dados de , ou seja, sem qualquer tipo de tipo de melhoria na cobertura mas uma piora na precisão. Com isto concluímos que é preferível manter o modelo inicial sem qualquer tipo de otimização do modelo GradientBoost.

3.3 Experiência 3 - Escolha das 2 variáveis

Para escolher quais as 2 melhores variáveis pensámos fazer 2 métodos:

- o primeiro foi um ciclo que em cada iteração, iria criar uma cópia do conjunto de variáveis, remover uma coluna na iteração e, em seguida, usando-a as variáveis sem essa coluna, treinar e testar um modelo de GradientBoostingClassifier, e por fim medir a cobertura. Ele então compara os diversos valores da cobertura e seleciona o melhor. Ao conjunto de dados principal removemos a variável associada ao melhor desempenho. Em seguida aplicamos este algoritmo múltiplas vezes até termos apenas duas variáveis. As avaliações são feitas com base na validação cruzada de forma a minimizar os enviesamento causados pela separação do conjunto de dados.

- o segundo é um algoritmo que faz combinações de todas as variáveis de tamanho 2, e em seguida testa e mede a cobertura para ver qual é o conjunto de 2 variáveis que tem um melhor desempenho de cobertura. As medições são feitas através de uma validação cruzada para conseguir minimizar o enviesamento causado pelos dados que estamos a usar.

Para os algoritmos as melhores variáveis e respetivos desempenhos foram para o teste foram:

Modelo	**Variável 1**	**Variável 2**	**Precisão**	**Cobertura**
Modelo 1	Id	Y2s2_complete	0.86	0.86
Modelo 2	Y2s2_complete	Y4s2_complete	0.69	0.91

Tabela 3: Avaliações dos modelos com 2 variáveis

Observando as avaliações dos modelos vemos que o melhor modelo é o Modelo 2 com uma cobertura de 0.91 e as melhores 2 variáveis são as variáveis Y2s2_complete e Y4s2_complete. Estas cumprem os requisitos deste modelo secundário pois temos 2 variáveis, apesar de não ter o mínimo de 70% de precisão e tem uma cobertura bastante alta. Com a otimização deste modelo poderemos chegar aos valores mínimos de 70%.

3.4 Experiência 4 - Otimização do Modelo com as duas variáveis selecionadas

De forma a melhorar ao máximo a cobertura com este modelo com apenas 2 variáveis, procurámos então fazer uma otimização com GridSearch e validação cruzada, então realizámos esta otimização com os parâmetros que já tínhamos visto acima que eram os mais importantes. Nesta conseguimos sim uma melhoria de 2% na cobertura mas uma perda de 26% na precisão, em relação ao modelo que continha todas as variáveis presentes. Com este modelo o mínimo de 70% de precisão foi cumprido e temos a melhor cobertura. Os parâmetros ótimo são:

Parameter	**Value**
learning_rate	0.1
max_depth	2
n_estimators	400

Tabela 4: Gradient Boosting Classifier parameters

Por fim o nosso modelo final, com duas variáveis é um modelo:

Métrica	**Valor**
Cobertura	91%
Precisão	70%

Tabela 5: Métricas do Gradient Boosting Classifier de 2 variáveis

4 Discussão de Resultados e Conclusões

Os dois modelos avaliados possuem diferentes níveis de precisão e cobertura. O primeiro modelo, que utiliza todas as variáveis, obteve uma precisão de 0.96 e uma cobertura de 0.89. Isso significa que ele é capaz de prever corretamente 0.96 das vezes e cobre 0.89 dos casos de risco de abandono de estudo. O segundo modelo, que utiliza apenas 2 variáveis, obteve uma precisão de 0.70 e uma cobertura de 0.91. Isso significa que ele é capaz de prever corretamente 0.70 das vezes e cobre 0.91 dos casos de risco de abandono de estudo.
Sendo que a precisão mede a capacidade do modelo de classificar corretamente os alunos que realmente abandonam os estudos (classe positiva). A cobertura mede a capacidade do modelo de encontrar todos os alunos que estão em risco de abandonar os estudos.
Considerando o objetivo principal de maximizar a cobertura e ter um mínimo de 0.70 de precisão, o segundo modelo, apesar de ter uma precisão menor, tem uma cobertura maior, o que o torna mais adequado para o problema em questão. No entanto, é importante notar que o primeiro modelo possui uma precisão maior, o que pode ser importante em outros contextos. A conclusão a partir deste trabalho é que é possível utilizar informações do histórico académico para prever alunos em risco de abandonar os estudos. A análise estatística dos dados, juntamente com os pacotes, possibilitaram a construção de um modelo com boa precisão e cobertura. Além disso, também foi possível propor um modelo alternativo simplificado, mostrando que é possível obter bons resultados com menos informações.

Appendices

	Y0s1_enrol	Y0s2_enrol	Y1s1_enrol	Y1s1_complete	Y1s1_grade
count	2110.0	2110.0	2110.0	2110.0	2110.0
mean	27.0	28.9	25.4	18.6	10.4
std	10.8	13.7	13.3	13.1	6.2
min	0.0	0.0	0.0	0.0	0.0
25%	24.0	24.0	23.0	2.0	10.0
50%	30.0	30.0	30.0	23.5	12.7
75%	33.0	36.0	34.0	30.0	14.7
max	66.0	93.0	90.0	90.0	18.2

	Y1s2_enrol	Y1s2_complete	Y1s2_grade	Y2s1_enrol	Y2s1_complete
count	2110.0	2110.0	2110.0	2110.0	2110.0
mean	27.7	19.9	10.8	22.8	16.2
std	14.2	13.4	6.3	15.7	13.1
min	0.0	0.0	0.0	0.0	0.0
25%	27.0	6.0	10.0	0.0	0.0
50%	30.0	24.0	13.0	30.0	19.0
75%	36.0	30.0	15.2	30.0	30.0
max	90.0	90.0	19.4	102.0	60.0

	Y2s1_grade	Y2s2_enrol	Y2s2_complete	Y2s2_grade	Y3s1_enrol
count	2110.0	2110.0	2110.0	2110.0	2110.0
mean	9.0	24.3	17.3	9.3	16.2
std	6.4	16.5	13.6	6.6	16.4
min	0.0	0.0	0.0	0.0	0.0
25%	0.0	0.0	0.0	0.0	0.0
50%	12.0	30.0	23.0	12.3	20.0
75%	13.9	34.0	30.0	14.4	30.0
max	20.0	89.0	52.0	18.5	75.0

	Y3s1_complete	Y3s1_grade	Y3s2_enrol	Y3s2_complete	Y3s2_grade
count	2110.0	2110.0	2110.0	2110.0	2110.0
mean	11.0	6.3	17.1	11.5	6.4
std	13.0	6.7	18.1	13.3	6.9
min	0.0	0.0	0.0	0.0	0.0
25%	0.0	0.0	0.0	0.0	0.0
50%	0.0	0.0	18.2	0.0	0.0
75%	26.0	13.0	30.0	27.0	13.5
max	60.0	18.0	141.0	49.0	18.0

	Y4s1_enrol	Y4s1_complete	Y4s1_grade	Y4s2_enrol	Y4s2_complete
count	2110.0	2110.0	2110.0	2110.0	2110.0
mean	6.5	3.2	2.3	7.1	3.4
std	13.3	7.8	4.9	14.9	8.3
min	0.0	0.0	0.0	0.0	0.0
25%	0.0	0.0	0.0	0.0	0.0
50%	0.0	0.0	0.0	0.0	0.0
75%	0.0	0.0	0.0	0.0	0.0
max	80.0	36.0	18.0	83.0	59.0

	Y4s2_grade	Rest_enrol	Rest_complete	Rest_grade
count	2110.0	2110.0	2110.0	2110.0
mean	2.2	11.7	3.3	1.0
std	4.8	49.9	14.6	3.4
min	0.0	0.0	0.0	0.0
25%	0.0	0.0	0.0	0.0
50%	0.0	0.0	0.0	0.0
75%	0.0	0.0	0.0	0.0
max	17.2	623.5	159.0	19.0

Tabela 6: Apêndices do dados descritivos dos conjuntos de dados