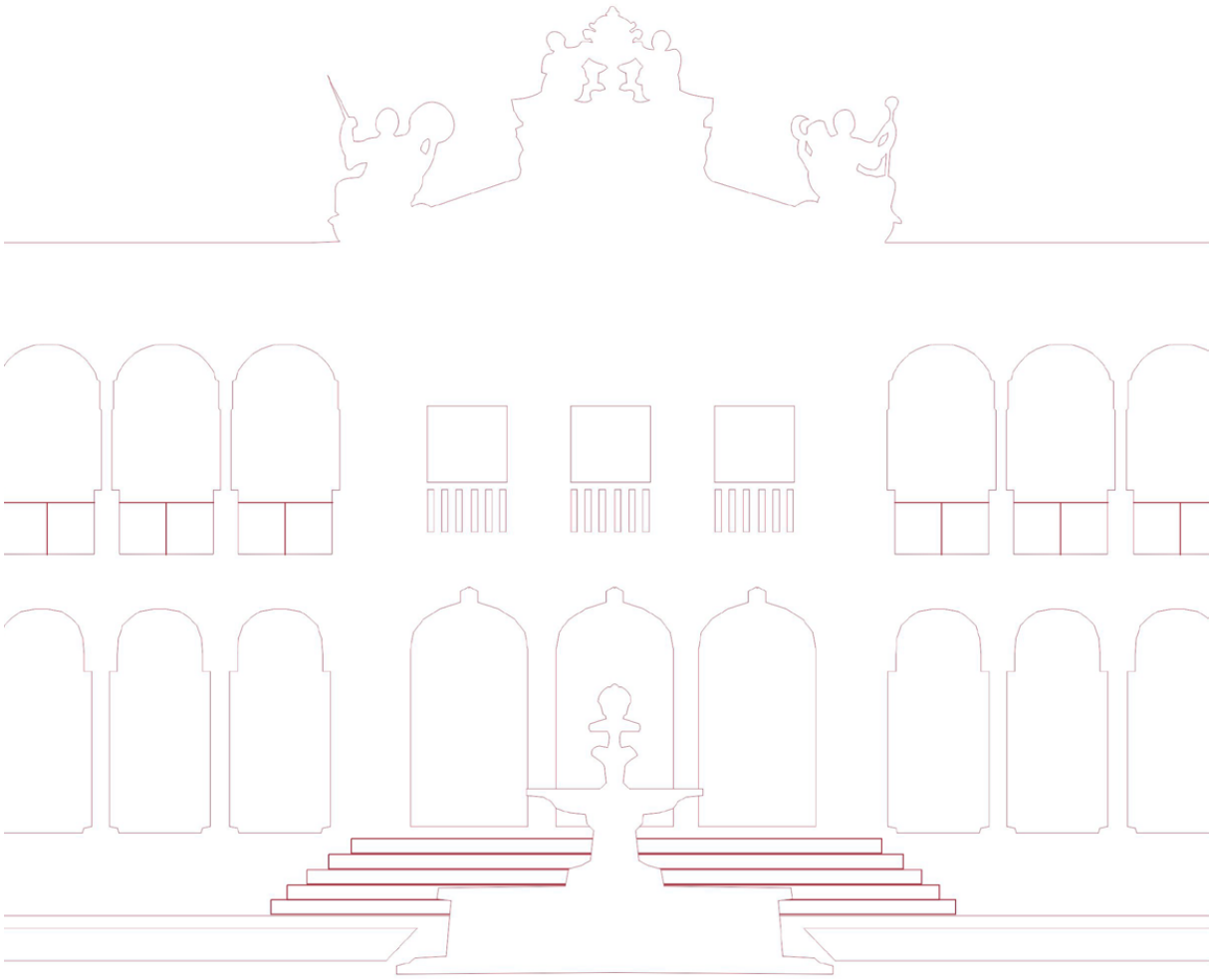
DataSet - Students Dropout

Licenciatura em Eng. Informática

Aprendizagem Automática

UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA



Helder Godinho 42741
Gonçalo Barradas 48402
Guilherme Grilo 48921

Docente: Luís Rato

Conteúdo

1 Introdução

Com o objetivo de avaliar os conhecimentos adquiridos na Unidade Curricular de Aprendizagem Automática, foi proposta a implementação de um modelo preditivo a fim de verificar quais os alunos em risco de abandonar os estudos, na linguagem Python. Recorrendo ao histórico académico de cada aluno obtém-se os parâmetros para avaliação, ou seja, o curso onde o aluno está inserido, os ECTs em que o aluno se encontra matriculado e os que já foram concluídos e ainda a média do próprio aluno. O objetivo da construção deste modelo preditivo é utilizar vários algoritmos, analisar os resultados de precisão e exatidão e, por fim, escolher o melhor algoritmo a partir dos resultados citados.

De forma a ser possível recorrer aos algoritmos de avaliação de dados sem ter que os implementar, foi necessário recorrer a uma biblioteca da linguagem Python, a scikit-learn. A scikit-learn é uma biblioteca de aprendizagem automática desenvolvida para a linguagem Python, que oferece vários algoritmos de classificação e regressão e que funciona bastante bem em conjunto com outras bibliotecas como o NumPy, utilizada para operações algébricas, em arrays e com o pandas. Pandas é uma biblioteca criada para a linguagem Python, utilizada para manipulação e análise de dados, isto é, o pandas oferece um vasto leque de estruturas e operações com o intuito de manipular conjuntos de dados.

Inicialmente foi necessário definir quais os algoritmos que seriam selecionados para testar o conjunto de dados e, devido à familiaridade e conhecimento, mesmo que reduzido, em relação a certos algoritmos, foram escolhidos seis algoritmos em específico para o efeito.

O algoritmo K-vizinhos, ou kNN, dado um conjunto de dados para teste, é escolhida uma instância de teste, e o algoritmo é responsável por encontrar os k vizinhos mais próximos dessa mesma instância, pois a classe positiva, isto é, a classe a que a instância classificada pertence, é dada pela classe que ocorre com maior frequência de entre os k vizinhos selecionados.

O algoritmo de Naive Bayes é um algoritmo de aprendizagem supervisionado que se baseia no teorema de Bayes e é utilizado sobretudo para resolver problemas de classificação. É ainda considerado um dos algoritmos de classificação mais simples e efetivo, o que ajuda na construção de rápidos modelos de aprendizagem automática que são responsáveis por realizar previsões rápidas. De maneira a explicar o que é o algoritmo de Naive Bayes, é necessário explicar o significado do nome do próprio algoritmo. Naive, pois assume que uma ocorrência de uma determinada característica é independente de outras ocorrências de outras características, ou seja, cada característica contribui para a identificação de todo o objeto sem depender umas das outras. Bayes, porque já na fase de aplicação do algoritmo, este baseia-se no teorema de Bayes, isto é, a fórmula do algoritmo é aplicada baseando-se no princípio da probabilidade condicionada.

Já o algoritmo k-folds é um método de validação cruzada que consiste basicamente em dividir um conjunto de dados em k subconjuntos. É escolhido um sub-conjunto, que é utilizado para teste e os restantes sub-conjuntos são usados para treino do modelo. Este processo é feito para todos os k sub-conjuntos, isto é, até que o conjunto de testes utilize cada um dos sub-conjuntos

No caso do algoritmo Random Forest é um método que cria, de forma aleatória, várias árvores de decisão, que consistem em estruturas de previsão mais simples que criam uma estrutura semelhante com uma árvore onde os ramos das mesmos são os caminhos que o algoritmo opta com o objetivo de chegar ao resultado final. De seguida, o algoritmo faz uma previsão de cada árvore e, por fim, combina o resultado de todas elas de maneira a atingir o resultado pretendido.

O algoritmo Extra Trees é similar ao algoritmo Random Forest, sendo que a diferença entre ambos está na aleatoriedade, pois o algoritmo Extra Tree introduz mais um fator de aleatoriedade nos seus processos, que consiste em separar os dados de forma aleatória, após a seleção dos mesmos para o primeiro nó da árvore.

Por último, foi utilizado ainda para classificação o algoritmo Gradient Boosting, que constrói um modelo de previsão a partir de modelos de previsão fracos, isto é, o algoritmo cria uma corrente de modelos fracos, onde cada um é ajustado e reduzido o erro do modelo anterior. Este tipo de iterações são repetidas até que a diferença entre o valor real e o valor previsto seja a mínima possível.

2 Algoritmo escolhido

Foram considerados vários pontos fundamentais na decisão do melhor algoritmo, a implementação e a maneira como funciona cada algoritmo, pois, atendendo ao conjunto de dados utilizado para testes, existem algoritmos mais apropriados do que outros. Os resultados de cobertura e precisão devem ser o mais próximo de 1, tendo sempre em atenção o ponto referido anteriormente, pois, por vezes, utilizar um algoritmo mais indicado para um conjunto de dados resulta numa avaliação mais eficiente do que basearmo-nos apenas nos resultados de cobertura e precisão. Também foi considerada a familiaridade com cada um dos algoritmos, isto é, existem algoritmos que foram lecionados nesta unidade curricular, o que aumenta o conhecimento em relação dos mesmos.

Após uma exaustiva análise e recorrendo aos pontos acima mencionados, a decisão final recaiu sobre o algoritmo Gradient Boosting pois, comparado com os restantes algoritmos utilizados, foi o que apresentou os resultados mais satisfatórios entre cobertura e precisão. Outro motivo pelo qual a escolha recaiu sobre o este algoritmo, foi o funcionamento do próprio, pois este, baseia-se na técnica de criar uma corrente de modelos fracos, sendo que a cada modelo testado, o mesmo é ajustado e melhorado em relação ao anterior e na opinião dos elementos do grupo, é o que apresenta resultados mais fidedignos realizando-se mais ajustes e testes.

3 Discussão de resultados

De forma a ser possível escolher um algoritmo, primeiramente houve a necessidade de testar os dados fornecidos em diferentes algoritmos. Recorre-se a uma análise dos seus resultados de forma a perceber qual o melhor algoritmo e o porquê do mesmo ser melhor para classificar este tipo de dados em relação aos demais. Após todos os testes feitos aos dados, isto é, após o conjunto de dados ser analisado e classificado por todos os algoritmos escolhidos, foram obtidos alguns resultados e são eles:

1. Tabela de resultados do algoritmo K-vizinhos

Nº de vizinhos	**Cobertura**	**Precisão**
1	0.80	0.85
3	0.76	0.86
5	0.69	0.82
10	0.44	0.87
15	0.40	0.80
20	0.33	0.80

2. Tabela de resultados do algoritmo Naive-Bayes

Cobertura	**Precisão**
0.96	0.30

3. Tabela de resultados do algoritmo k-Folds

Cobertura	**Precisão**
0.89	0.98

4. Tabela de resultados do algoritmo Random Forest

Cobertura	**Precisão**
0.87	0.98

5. Tabela de resultados do algoritmo Extra Trees

Cobertura	**Precisão**
0.87	0.84

6. Tabela de resultados do algoritmo classificador Gradient Boosting

Cobertura	**Precisão**
0.91	0.98

Após a análise dos resultados apresentados na secção 3, foi possível verificar que, à partida, existem alguns algoritmos que podem ser imediatamente descartados, pois apresentam resultados insatisfatórios. O algoritmo K-vizinhos, devido à sua funcionalidade, apresenta resultados demasiadamente dispersos na cobertura, o que torna o algoritmo pouco eficiente na sua avaliação final; o algoritmo de Naive-Bayes apresenta uma precisão demasiado baixa para entrar na equação; e ainda o algoritmo Extra Trees, apesar de apresentar resultados bastantes satisfatórios, quando comparado com os restantes algoritmos, encontra-se ligeiramente abaixo.

Com os três algoritmos que sobraram, foi necessário efetuar mais alguma pesquisa sobre o funcionamento dos algoritmos, de forma a adquirir mais conhecimento e perceção acerca dos mesmos. Após uma análise exaustiva, chegou-se à conclusão que o melhor algoritmo seria o Gradient Boosting, devidos aos motivos explicados nas secções 2 e 3.

4 Conclusão

O algoritmo escolhido, o Gradient Boosting, demonstrou ser mais eficiente que os restantes no conjunto de dados testado, pois apresentou melhores resultados que os restantes algoritmos. Apesar do algoritmo de k-Folds e do algoritmo Random Forest também apresentarem resultados bastante positivos e, por essa razão, serem também opções válidas de escolha, a diferença no resultado da cobertura e até mesmo a forma como estes dois algoritmos classificam o conjunto de dados quando comparado com o Gradient Boosting, levou a que a decisão recaísse sobre o último. Sendo que, o Gradient Boosting se baseia na técnica de criação de modelos fracos e efetua ajustes em cada um destes com o objetivo de diminuir para o mínimo a diferença entre o valor real e previsto, fez toda a diferença na escolha, pois foi possível concluir que este realiza mais testes que muitos dos algoritmos utilizados, o que leva a resultados mais fidedignos e confiáveis, e em Machine Learning esse é um fator muito importante e bastante apreciado.

Referências

[Harris et al., 2020] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., and et al. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

[pandas development team, 2020] pandas development team, T. (2020). pandas-dev/pandas: Pandas.

[Rato, 2022] Rato, L. (2022). Aulas de aprendizagem automática. *in Universidade de Évora*.