Relatório Final do Projeto

Victor Hugo Rodrigues da Cunha - vhrc@cin.ufpe.br

1. Introdução

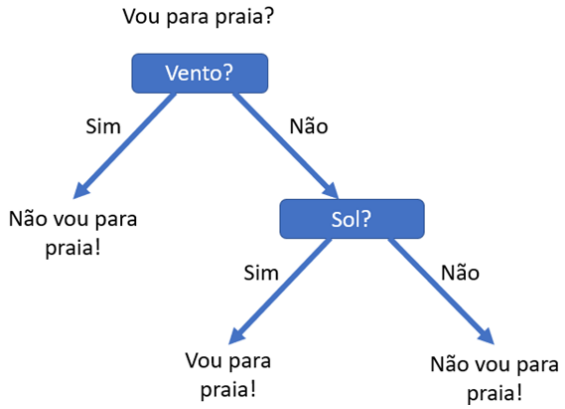
O dataset utilizado se encontra disponível no Kaggle através do link https://www.kaggle.com/spscientist/students-performance-in-exams, ele é formado por notas de estudantes com seu desempenho em três áreas: cáuculo, leitura e escrita. Além disso, ele contém também dados sociais dos alunos: grupo étnico, gênero, escolaridade parental e auxílio alimentação (almoço com valor reduzido ou não) e se foi realizada a prova de preparação.

O objetivo foi utilizar modelos de machine learning para avaliar a possibilidade de prever o desempenho dos alunos em sua nota para escrita, considerando as features de nota em cálculo, nota em leitura, realização de curso de preparação para prova, auxílio no almoço e nível de escolaridade parental (etnia e gênero foram desconsiderados), além de entender o quanto cada fator influencia na previsão da nota final.

2. Fundamentos

a. Árvore de Decisão

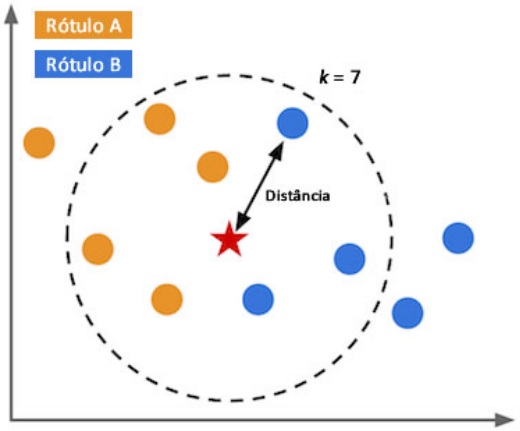
Algoritmo de aprendizagem de máquina que cria uma estrutura com vários pontos de decisão, organizando-os em nós. A construção da árvore de decisão é baseada nos dados de treino, através deles serão determinados o conteúdo e as regras para os nós e os ramos. O caminho de decisão formado na árvore permite que uma previsão seja feita com dados semelhantes.



Exemplo de árvore de decisão (fonte: didatica.tech)

b. K Vizinhos Próximos

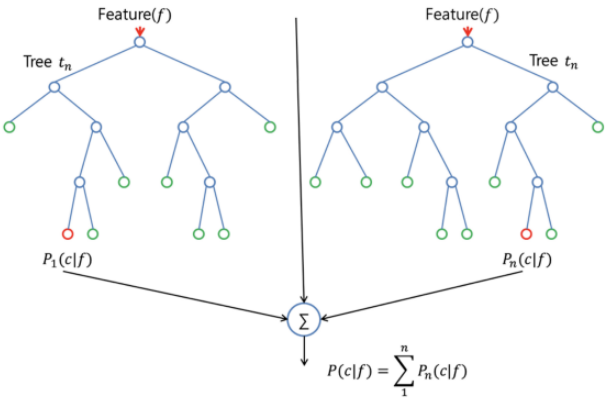
Algoritmo que utiliza das proximidades entre dados para estabelecer previsões. O algoritmo calcula a distância entre os dados utilizando uma métrica de distância, a distância euclidiana, por exemplo, e considera os k (valor determinado) vizinhos mais próximos para determinar a que classe pertenceria os dados. Por exemplo, se o valor de k for 7, serão considerados os 7 elementos mais próximos, e a partir de seu valor, será determinado o rótulo do dado buscado.



Representação visual do funcionamento do K Vizinhos Próximos (fonte: computacaointeligente.com.br)

c. Floresta Aleatória

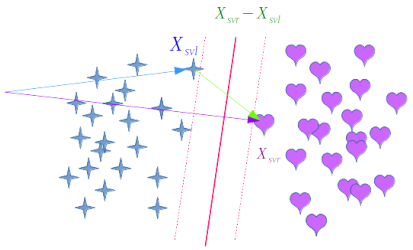
Semelhante à árvore de decisão, só que dessa vez fazendo a criação de diversas árvores diferentes, e utilizando parâmetros aleatórios na composição de seus nós de decisão. Ela atua com subconjuntos aleatórios de suas features e dependendo da quantidade de árvores geradas acaba por necessitar de um poder computacional maior.



Representação de floresta aleatória com duas árvores (fonte: medium.com/machina-sapiens)

d. Máquinas de Vetor de Suporte

O algoritmo busca a melhor forma de dividir o conjunto de dados através de uma fronteira. A divisão pode ser feita de forma linear ou não-linear. Através dessa divisão será possível realizar previsões.



Representação visual de um algoritmo de máquinas de vetor de suporte (fonte: https://towardsdatascience.com/)

3. Metodologia

A metodologia pode ser organizadas em passos:

- Importação das bibliotecas necessárias.

Dados

- Importação do dataset;
- Exclusão das colunas do dataset com features que não serão usadas (gender e race/ethnicity);
- Ajuste na coluna a ser usada como classe: writing score
  - Originalmente trazia valores numéricos entre 19 e 100;
  - Foi organizado em 4 classes
    - valores < 40 **=>** classe 1
    - valores >= 40 e < 60 **=>** classe 2
    - valores >= 60 e < 80 **=>** classe 3
    - valores >= 80 **=>** classe 4
- As demais features foram convertidas manualmente em valores numéricos;
- Através do train_test_split foi feita a divisão entre dados de teste e dados de treino, onde 30% dos dados disponíveis foram usados para treino.

Algoritmos

Árvore de Decisão

- O primeiro algoritmo utilizado foi a árvore de decisão, gerada sem nenhum parâmetro especificado, juntamente com sua acurácia e importância das features;
- Para uma melhor visualização foi usado o graphviz para renderizar a árvore;
- Com o interactive foi gerada ainda uma árvore de decisão interativa, o que permitia que a árvore fosse visualizada graficamente ao mesmo tempo em que seus parâmetros poderiam ser modificados, contribuindo para a geração de várias árvores.

k-vizinhos mais próximos

- O segundo algoritmo foi o KNN (k-vizinhos mais próximos), gerado com o k = 3;
- Após isso ele foi executado mais vezes, para k = 1, k = 3, k = 5, k = 7 e k = 9
- Em todas as execuções foi exposto o respectivo valor de acurácia;

Florestas Aleatórias

- Primeiramente foi gerada uma floresta aleatória com 100 árvores, com acurácia e importância das features;
- Depois, mais florestas foram geradas, com suas respectivas acurácias e importância das features. as florestas tiveram 100, 200, 500, 800, 1000, 2000 e 3000 árvores.

Máquinas de Vetor de Suporte

- A primeira máquina de vetor de suporte usou o kernel linear, sua acurácia foi exposta;
- Após isso, mais máquinas de vetor de suporte foram executadas, calculando sua acurácia para os seguintes kernels: linear, poly e rbf.

Remoção de Features

- Verificando os resultados ficou claro que algumas features eram muito menos importantes que outras, e que seria interessante executar os algoritmos desconsiderando elas;
- Dentre as menos importantes, as features de lunch e de test preparation course foram as escolhidas para serem descartadas;
- As colunas contendo essas features foram removidas do dataset, os dados foram novamente divididos (seguindo a mesma proporção), e todos os algoritmos foram novamente executados (sem nenhuma modificação neles próprios).

4. Resultados

Árvore de Decisão

i. Árvore de Decisão sem parâmetros

Árvore gerada usando os parâmetros padrões, sem especificar profundidade, largura etc.

A tabela mostra a importância de cada feature no modelo e a acurácia da árvore:

parental level of education	lunch	test preparation course	math score	reading score	**acurácia**
0.06789059178626686	0.016889444513026368	0.027941713253265766	0.13699429045845057	0.7502839599889903	0.7433333333333333

A acurácia foi de 74%, e as features que menos contribuíram foram lunch e test preparation course, ambas com menos de 0.03.

ii. Árvores de Decisão com parâmetros especificados

A tabela mostra os parâmetros especificados, a importância de cada feature no modelo e a acurácia final da árvore:

critério	**divisor**	**profundidade**	parental level of education	lunch	test preparation course	math score	reading score	**acurácia**
gini	best	5	0.008396437710832773	0.005362822719982697	0.026299121207470551	0.049500332981131711	0.910441285380582 3	0.823333333333334
entropy	best	5	0.017289655427641867	0.003089275347318822	0.027302556758756733	0.037847596697159344	0.914470915769123 3	0.826666666666667
gini	random	5	0.019134531113663648	0.001491815593459511 4	0.035170471363175235	0.173554001943996 48	0.770649179985705 1	0.82
entropy	random	10	0.069424093980886 77	0.024704033268328433	0.042860278910506 2	0.058588017629885 53	0.804423576210393 2	0.8

Com menor profundidade a acurácia obteve melhores resultados, e o valor de reading score se destacou em nível de importância em todas as execuções.

k-vizinhos mais próximos

A tabela mostra a acurácia conseguida para cada valor de K:

Valor de K	**Acurácia**
1	0.76
3	0.8
5	0.8033333333333333
7	0.81
9	0.81

Florestas Aleatórias

A tabela mostra os resultados obtidos com florestas aleatórias de acordo com a quantidade de árvores:

árvores	parental level of education	lunch	test preparation course	math score	reading score	**acurácia**
100	0.065993468893	0.021836390438	0.027249792143	0.258388551380	0.626531797143	0.826666666666

	75338	25492	678314	4315	8819	6667
200	0.06905 0734818 66016	0.02168 1059624 27665	0.02830 0107712 70565	0.26023 0453092 09886	0.62073 7644752 2588	0.80333 3333333 3333
500	0.06841 2892050 66627	0.02127 0612907 770832	0.02826 6644827 462252	0.26362 1061873 93486	0.61842 8788340 1658	0.80666 6666666 6666
800	0.06949 0150436 89459	0.02071 2676789 929396	0.02864 8668845 465516	0.26087 3717685 6628	0.62027 4786242 0477	0.80333 3333333 3333
1000	0.06965 9378857 43215	0.02090 5485624 737537	0.02879 9431523 639795	0.26378 9072701 525	0.61684 6631292 6654	0.81
2000	0.06986 5479675 7258	0.02082 2499929 430395	0.02887 3017657 358963	0.26488 1241211 40455	0.61555 7761526 0802	0.80666 6666666 6666
3000	0.06930 0104102 79687	0.02110 1681923 825347	0.02894 9005745 23716	0.26324 0647167 8833	0.61740 8561060 2573	0.80666 6666666 6666

Máquinas de Vetor de Suporte

A tabela mostra a acurácia das MVS de acordo com seu kernel:

kernel	**acurácia**
linear	0.86
poly	0.84
rbf	0.8333333333333334

5. Resultados Removendo Features

Possuindo menores valores no nível de importância, os algoritmos foram executados novamente, mas dessa vez sem as features de lunch e test preparation course.

Árvore de Decisão

i. Árvore de Decisão sem parâmetros
   Árvore gerada usando os parâmetros padrões, sem especificar profundidade, largura etc.

A tabela mostra a importância de cada feature no modelo e a acurácia da árvore:

parental level of education	math score	reading score	**acurácia**
0.08103160697959019	0.18212214618645972	0.7368462468339501	0.8066666666666666

ii. Árvores de Decisão com parâmetros especificados

A tabela mostra os parâmetros especificados, a importância de cada feature no modelo e a acurácia final da árvore:

critério	**divisor**	**profundidade**	parental level of education	math score	reading score	**acurácia**
gini	best	5	0.02904395155707 6075	0.04668012769064 1234	0.91044128538058 23	0.8233333333333334
entropy	best	5	0.01535263239179 4708	0.05101645438218 266	0.93363091322602 27	0.8266666666666667
gini	random	5	0.00780885787452 2605	0.00793823207038 2937	0.98425291005509 44	0.8733333333333333
entropy	random	10	0.04849244835114 6224	0.06682252407132 151	0.88468502757753 22	0.8266666666666667

k-vizinhos mais próximos

A tabela mostra a acurácia conseguida para cada valor de K:

Valor de K	**Acurácia**
1	0.8033333333333333
3	0.8433333333333334
5	0.84
7	0.8433333333333334
9	0.8566666666666667

Florestas Aleatórias

A tabela mostra os resultados obtidos com florestas aleatórias de acordo com a quantidade de árvores:

árvores	parental level of education	math score	reading score	**acurácia**
100	0.060785359 06828809	0.331979484 7981852	0.607235156 1335266	0.836666666 6666667
200	0.063166920 40421984	0.320314862 88127686	0.616518216 7145032	0.83
500	0.062225346 69952118	0.325163083 6602394	0.612611569 6402393	0.833333333 3333334
800	0.064324806 03005256	0.322328407 6080934	0.613346786 361854	0.83
1000	0.062318561 636242054	0.320852379 8239583	0.616829058 5397995	0.836666666 6666667
2000	0.062440064 78335014	0.321859156 6039556	0.615700778 6126943	0.833333333 3333334
3000	0.063153890 659692	0.322846041 87130234	0.614000067 4690057	0.836666666 6666667

Máquinas de Vetor de Suporte

A tabela mostra a acurácia das MVS de acordo com seu kernel:

kernel	**acurácia**
linear	0.87
poly	0.8733333333333333
rbf	0.8633333333333333

6. Conclusão

De modo geral os algoritmos obtiveram bons resultados, quase sempre ultrapassando a marca de 80% de acurácia, mesmo com mudanças em seus respectivos parâmetros. Dentre todos, o algoritmo que obteve o pior resultado foi a Árvore de Decisão quando utilizada sem especificação de nenhum critério, ela conseguiu apenas 74% de acurácia. Isso mostra que ela estava tão adaptada aos dados de treino que não conseguiu ter um bom desempenho na hora do teste. A poda acabou por ser essencial para que o seu resultado começasse a ultrapassar os 80% de acurácia.

Outro ponto interessante é com relação às features utilizadas, quando todas as cinco foram usadas em conjunto o nível de importância costumou a mostrar valores mais baixos para as features de lunch e test preparation course. Executando novamente todos os algoritmos, a falta dessas duas features contribuiu para melhores resultados nas acurácias, com, inclusive, o algoritmo de Máquinas de Vetor de Suporte alcançando 87% de acurácia em sua versão com kernel polynomial.

Os resultados finais trouxeram a confirmação de que o desempenho em leitura (reading score) está bem relacionado ao desempenho em escrita (writing score, que foi utilizada como classe), e que o acesso ao almoço e o curso de preparação para a prova são fatores que influenciam menos na nota final de escrita, chegando a atrapalhar a previsão.

A execução dos algoritmos, além de tudo, demonstrou a importância da boa escolha de features, já que o uso de features limitadas às mais importantes trouxe melhores resultados que o uso de todas as features. Por fim, ficou claro também o quão vantajoso diferentes algoritmos para se buscar uma melhor qualidade na previsão. Uma proposta de trabalho futuro seria uma nova remoção de feature menos importante, nesse caso a parental level of education para focar a previsão apenas entre as diferentes notas nas áreas de cálculo, leitura e escrita.