Aprendizado de máquina e inteligência artificial em física 4305512

Bruno Penteado Monteiro - NºUSP 10300723

22 de Abril de 2024

Atividade 2

1 Análise exploratória

Inicialmente os dados foram importados do arquivo *.csv* para o pandas em um notebook python. Foram verificados uma grande quantidade de dados faltantes, que devem ser preenchidos ou excluídos da análise sob justificativa plausível. Sendo o objetivo da análise criar modelos capazes de prever as notas de matemática, e as notas de matemática possuírem 8173 dados faltantes, foi criado um novo DataFrame removendo todas as linhas sem notas de matemática. Essa decisão foi tomada visto que as linhas sem a variável alvo não podem compor o conjunto de dados de treino e teste. No fim da análise, o objetivo será prever as notas de matemáticas removidas neste passo. Adicionalmente, foi verificado que após a remoção das notas de matemática nulas, houve um remanescente de 36 notas nulas de ciências da natureza, ciências humanas e linguagens e códigos. Visto a baixa estatística desse conjunto de dados perto dos mais de 10100 dados remanescentes, essas linhas também foram removidas da mesma forma.

Verificando a nova base de dados temos apenas mais 6469 dados faltantes na coluna TP_ENSINO, referentes à alunos que não responderam a essa alternativa. Para solucionar essa questão foi atribuído um valor 0, criando uma nova categoria "não respondido"a essa característica, assim como foi feito em outras perguntas.

Com o objetivo de remover os atributos categóricos sem grande importância para a análise proposta foi encontrado redundâncias nos atributos TP_STATUS_REDACAO, IN_TREINEIRO e TP_PRESENCA. No caso do TP_STATUS_REDACAO, foi verificado que 97,7% da base de dados possui resultado 1, o que significa que a redação não possui problemas. Todos os outros resultados possuem individualmente uma estatística menor que 1% da base de dados, e implicam em uma nota 0 de redação, informação que já está presente em NU_NOTA_REDACAO. Por conta desses dois fatores essa coluna foi excluída do DataFrame de treino e teste. No caso da coluna IN TREINEIRO, que possui como resposta 1 para treineiro e 0 para não treineiro, foi verificado que 100% das respostas 1 também responderam 3 para TP_ST_CONCLUSAO ("Estou cursando e concluirei o Ensino Médio após 2016") e 0 para TP_ANO_CONCLUIU ("Não informado", ou seja não concluiu). Dessa forma, essa coluna também foi considerada redundante para a análise e foi removida do DataFrame de treino e teste. Por fim, as variáveis TP_PRESENCA também possuem valor 1 ou 0 para presente ou não, e foram removidas já que no caso de não presente a respectiva nota é zerada.

Através da abordagem descrita, foi possível criar um DataFrame ideal para as etapas de treino e teste, que é apresentado na tabela 1.1. Além disso, também é apresentado um heatmap, que nos mostra a correlação entre as características dos dados, na figura **??**.

2 Regressão multilinear

Para realizar as etapas de treino e teste, o conjunto de dados tratado, conforme descrito na sessão 1, foi dividido em 2, sendo 40% das linhas o conjunto de treino, e 60% o conjunto de teste. Inicialmente foi treinado um modelo mais simples que utiliza apenas as categorias com maior correlação em relação à nota de matemática, como pode ser visto em 1.1. Essas características representam as notas dos alunos nas outras modalidades, que são valores discretos, e todas obtiveram um módulo de correlação

Tabela 1.1: Dataframe de dados preparados para a etapa de treino e validação.

TP_ESCOLA	IDADE	ST_CONC	ANO_CONC	TP_ENSINO	NOTA_CN	NOTA_CH	NOTA_LC	NOTA_MT	NOTA_RED
1	24	1	4	1.0	436.3	495.4	581.2	399.4	520.0
1	17	2	0	1.0	474.5	544.1	599.0	459.8	580.0
1	18	1	1	1.0	439.7	583.2	410.9	364.5	620.0
...	...	...	...	...	...	...	...	...	...
1	15	3	0	1.0	460.5	528.9	569.3	398.0	600.0
1	36	4	0	1.0	422.5	621.7	569.0	386.6	460.0
1	17	2	0	1.0	488.7	575.3	565.9	428.9	520.0

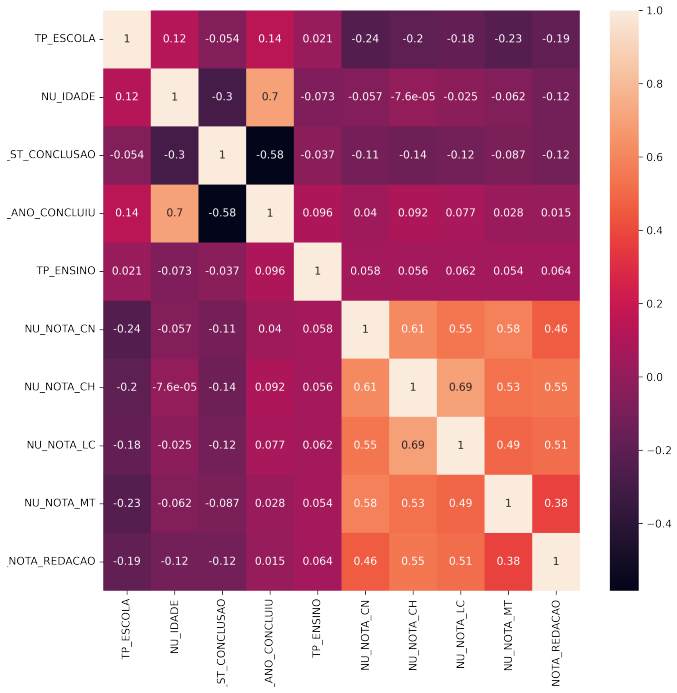

Figura 1.1: Matriz de correlações entre as características dos dados.

foi superior a 0,3. O modelo foi treinado e utilizado para avaliar tanto os conjuntos treino quanto teste. Os resíduos desse modelo podem ser vistos na figura 2.1.

Além do modelo utilizando apenas as notas, podemos treinar um modelo utilizando todas as variáveis da tabela 1.1, para assim podermos comparar os resultados desses dois modelos. Os resíduos desse modelo são apresentados na figura 2.2.

Podemos comparar os parâmetros obtidos com cada modelo através da figura 2.3. Nela, observamos que apesar da correlação dos parâmetros de nota serem superiores, os coeficientes das características TP_ESCOLA e TP_ENSINO possuíram um módulo maior quando ajustados todos simultaneamente. Esse efeito pode estar relacionado a uma multicolinearidade entre esses parâmetros, e por conta disso foi feito o teste de um modelo Lasso, para verificar se esses coeficientes diminuem ou são zerados. Visto que já foi observado que apenas as características de nota são suficientes para descrever as notas de matemática com um desempenho tão bom quanto todas as características, foi utilizado um valor de λ=5, com objetivo de diminuir bastante as dimensões do modelo. Os coeficientes Lasso são apresentados na figura 2.4.

A figura 2.4 nos mostra que alguns coeficientes que ficaram com valor absoluto grande no modelo multilinear foram zerados pelo método Lasso. Isso indica que havia uma multicolinearidade entre essas características que superestimava os coeficientes no modelo multilinear. As características remanescentes foram a da idade dos alunos junto com os 4 tipos de nota nas outras modalidades do

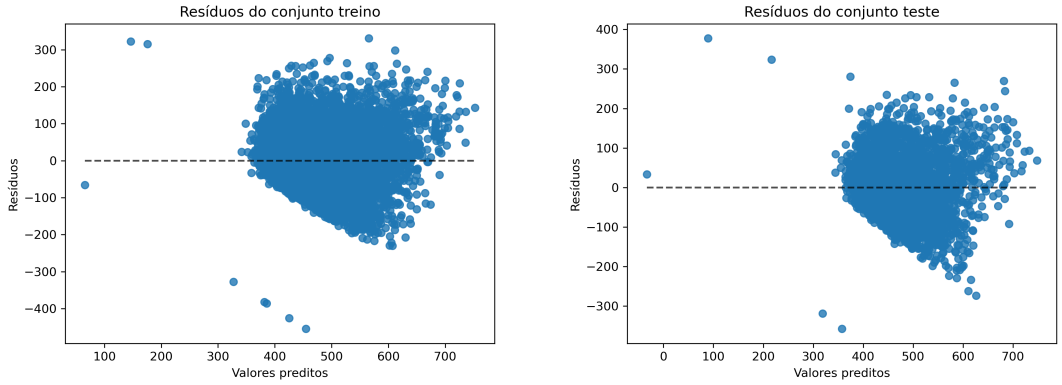

Figura 2.1: Resíduos do modelo de regressão multilinear utilizando apenas as notas dos alunos nas outras modalidades para o conjunto de treino (esquerda) e de teste (direita).

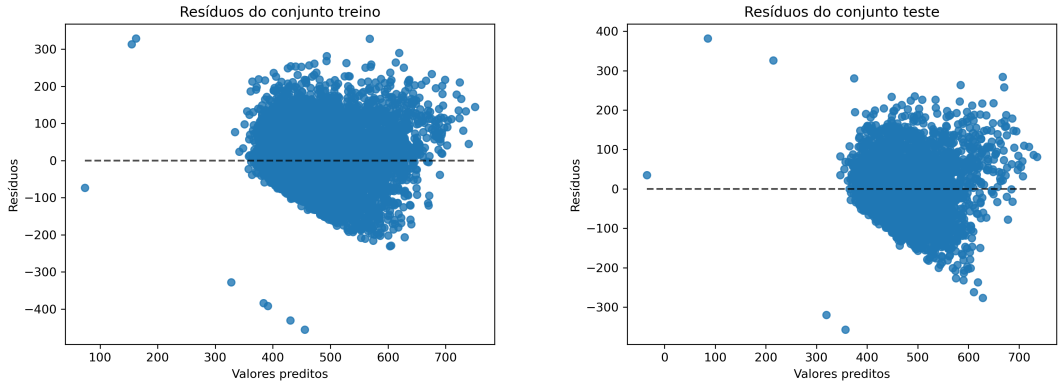

Figura 2.2: Resíduos do modelo de regressão multilinear utilizando todas as características para o conjunto de treino (esquerda) e de teste (direita).

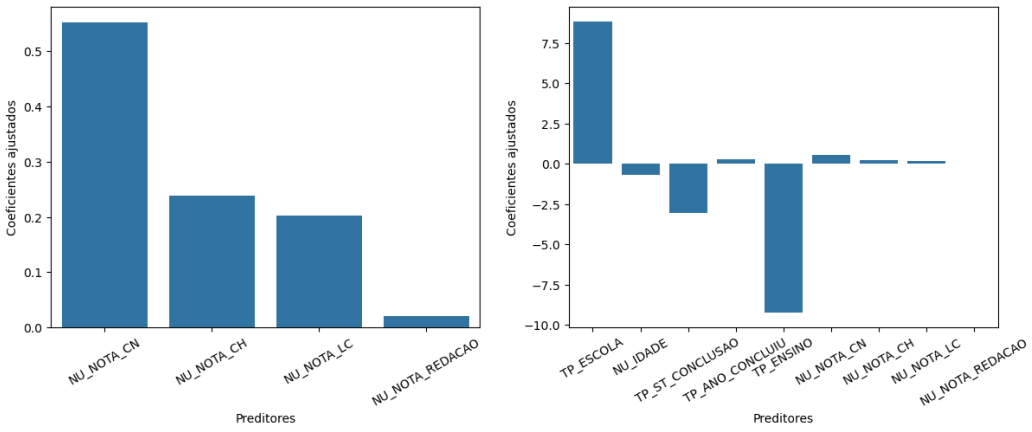

Figura 2.3: Parâmetros obtidos para o modelo de regressão linear utilizando apenas as notas (esquerda) e utilizando todos os atributos (direita).

ENEM.

Para comparar o desempenho desses modelos é disposto uma tabela com os valores de R^2, MSE e MAE para os conjuntos de treino e teste de cada modelo. A tabela de resultados dos desempenhos está disposta em 2.

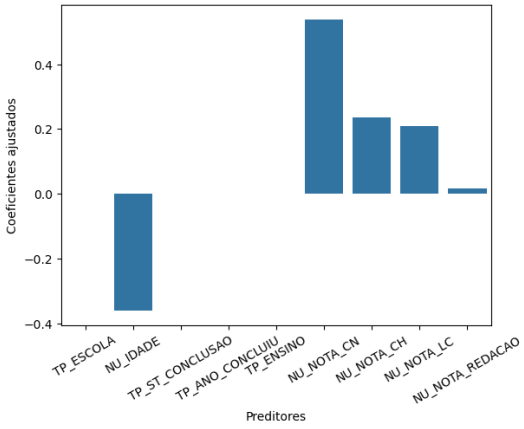

Figura 2.4: Parâmetros obtidos para o modelo de regressão linear Lasso.

	Notas (treino)	Notas (teste)	Todos (treino)	Todos (teste)	Lasso (treino)	Lasso (teste)
R^2	0.40	0.39	0.41	0.40	0.40	0.41
MSE	5917	6032	5888	6002	5917	5978
MAE	60.8	61.7	60.7	61.5	60.7	61.6

Tabela 2.1: Tabela de critérios de desempenho do modelo de regressão multilenear utilizando apenas as notas dos alunos e utilizando todas as características para os conjuntos de treino e teste.

Da tabela 2 verificamos que os resultados das métricas de desempenho ficaram bem parecidas entre os conjuntos de treino e de teste, que é um indicativo de que não houve overfiting. Os valores de R^2 ficaram em torno de 0,4, que mostra que os modelos possuem certa limitação em relação a prever as notas de matemática, ficando inclusive com um erro médio absoluto de cerca de 60 pontos. Todos os modelos obtiveram resultados de desempenho praticamente idênticos, entretanto, devemos considerar que o modelo que utiliza apenas as notas chegou nesses resultados utilizando menos características, o que pode ser interessante quando pensamos em limitações de processamento. O modelo Lasso representou o melhor modelo para o conjunto teste, mostrando que a idade dos alunos também é razoavelmente relevante além das notas das outras modalidades.

3 Random forest

A primeira abordagem para esse método, de forma semelhante à regressão multilinear, foi criar um modelo simplificado considerando apenas as notas dos alunos nas outras modalidades. Para isso foi realizada a mesma divisão do conjunto de dados em 40% para treino, e 60% para teste. Nesse modelo foi utilizado as configurações padrão da função RandomForestRegressor() do Sklearn. Os resultados de desempenho desse modelo para o conjunto treino e conjunto teste estsão dispostos na tabela 3.

	Notas (treino)	Notas (teste)
R^2	0.91	0.41
MSE	847	5985
MAE	22.7	61.1

Tabela 3.1: Métricas de desempenho do modelo de random forest padrão para os conjuntos de treino e teste.

Analisando os resultados de desempenho do modelo criado, tabela 3, é possível verificar uma grande diferença de desempenho entre os conjuntos de treino e de teste. O modelo ficou extremamente especializado em identificar as notas de matemática do conjunto treino, ficando com um R^2 de 0,91. Já no conjunto teste, esse mesmo resultado foi de 0,41, bem inferior. Isso é um indicativo de overfiting, se mostrando necessário mudar os parâmetros padrões do modelo de random forest do sklearn para

que este não fique tão especializado no conjunto treino.

Para verificar os parâmetros do random forest que impeçam o overfiting, foi utilizado o algoritmo GridSearchCV, também do Sklearn. Esse algoritmo realiza uma busca do melhor ajuste variando os parâmetros do modelo escolhido em formato de grid. De acordo com a página de documentação do random forest, os principais parâmetros que influenciam no comprimento da árvore são o max_depth e o n_estimators. Esses parâmetros foram sendo variados em grid para valores maiores dos valores padrão (max_depth=1, n_estimators=100). Após algumas iterações do algoritmo foram encontrados os valores max_depth=5 e n_estimators=550 com melhor métrica de desempenho.

Utilizando os parâmetros comentados no parâgrafo anterior, foram treinados dois modelos de random forest, sendo o primeiro utilizando apenas os valores de notas e idade (conforme visto na regressão multilinear) e o segundo utilizando todas as características da tabela 1.1. O resultado dos resíduos desses modelos estão dispostos na figura 3.1. Já os valores de desempenho de ambos os modelos são apresentados na tabela 3.

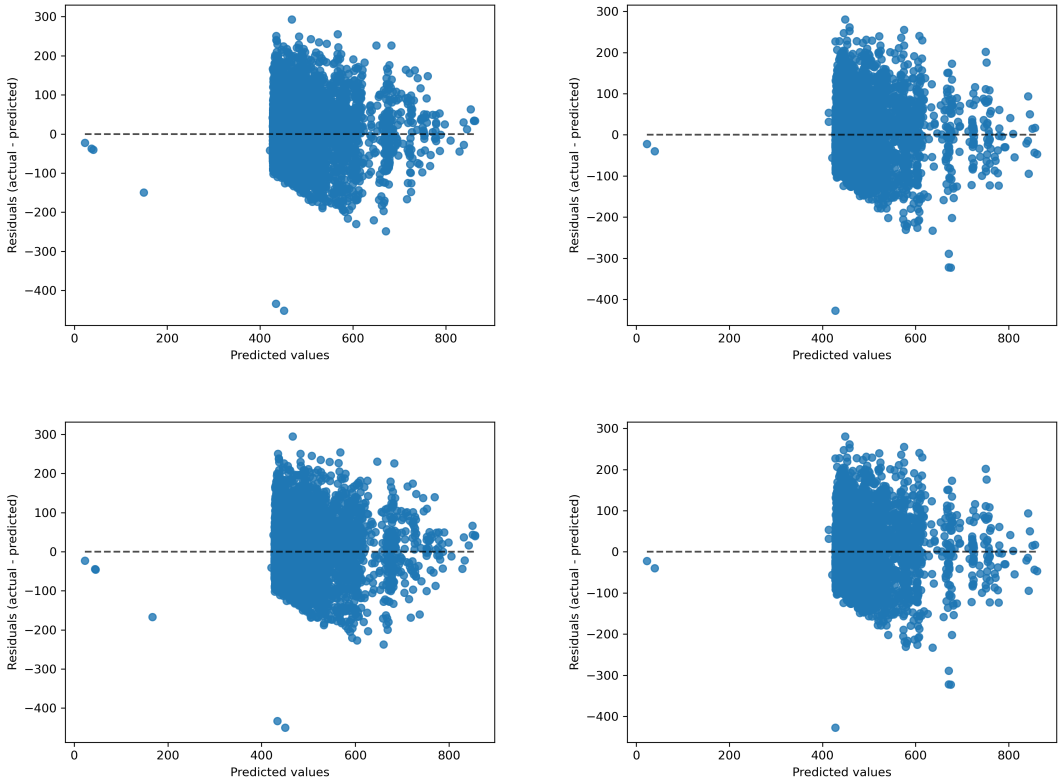

Figura 3.1: Resíduos do modelo de random forest utilizando apenas as notas dos alunos nas outras modalidades (cima) e utilizando todas as características (baixo) para o conjunto de treino (esquerda) e de teste (direita).

	Notas (treino)	Notas (teste)	Todos (treino)	Todos (teste)
R^2	0.47	0.45	0.48	0.45
MSE	5144	5553	5121	5550
MAE	56.9	59.1	56.8	59.1

Tabela 3.2: Métricas de desempenho do modelo de random forest para os conjuntos de treino e teste obtidas para os modelos Notas: notas nas outras modalidades e idade do aluno e Todos: utilizando todas as características.

Diante dos resultados de desempenho mostrados na tabela 3, podemos observar que agora a eficiência das predições de treino e teste estão mais similares, indicando que os modelos estão pre-

vendo melhor dados fora do conjunto treino. Os resultados de desempenho do modelo utilizando todos os parâmetros ficou ligeiramente melhor, entretanto, o resultado do modelo de notas+idade também obteve um desempenho próximo utilizando menos parâmetros.

4 Conclusões finais

Os modelos de regressão multilinear obtiveram resultados de desempenho bem próximos, com R^2 próximo de 0,4 e um MSE de cerca de 6000. O modelo de random forest com melhor desempenho utilizou todas as características da base de dados (após a redução) e obteve um R^2 de 0,45 e um MSE de 5550 para o conjunto de teste. Dessa forma, o random forest se mostrou ser um pouco mais adequado para lidar com a predição das notas de matemática do ENEM, dado a base de dados utilizada.

Os modelos treinados e analisados não obtiveram um desempenho muito satisfatório na predição de notas de matemática devido ao R^2 inferior a 0,5 e um erro absoluto médio de cerca de 60 pontos. Um possível motivo para isso é que os dados apresentados podem ter um alto ruído, acarretando assim em um desvio dos valores previstos. Outros motivos podem vir da baixa estatística de dados, ou da forma como a redução de dados foi feita.

Com a redução de dados optou-se por remover algumas variáveis categóricas que foram julgadas como redundantes. Dado que na regressão multilinear a única característica que se mostrou relevante além das notas foi a idade, que também é uma variável discreta, pode-se concluir que nesse caso as variáveis categóricas não estavam contribuindo para o modelo. Uma outra abordagem possível seria transformar as variáveis categóricas em variáveis binárias, para verificar se isso possibilitaria um impacto positivo no modelo, aumentando seu desempenho.