Nanodegree Engenheiro de Machine Learning

Projeto final

Felipe Alcino Luiz
20 de janeiro de 2018

I. Definição

Visão geral do projeto

O escopo de análise desse projeto é tentar entender e traçar a correlação entre os atributos do vinho que podem impactar na qualidade final do mesmo. Os dados utilizados aqui são originados de https://archive.ics.uci.edu/ml/datasets/wine+quality. Neste caso utilizaremos a base de dados relacionada aos vinhos vermelhos (red wine). O dataset é proveniente do "Vinho Verde" de Portugal. Dados sobre os tipos de uva e outras caracteristicas não foram utilizadas por questões de privacidade, utilizando apenas os valores finais, propriedades fisico-quimicas e sensoriais (como sabor, cheiro e etc.)

Descrição do problema

A questão do problema é correlacionar como que os atributos do vinho influenciam na sua qualidade final gerando um modelo que possa prever isso automaticamente.

Métricas

A performance é avaliada baseada na pontuação de 3 algoritmos usados na parte separada para teste no dataset. Utilizamos nesse caso F1 score que utiliza uma analise binária (onde 0 é o pior score e 1 é o melhor) baseada em quão bem o modelo consegue categorizar corretamente valores verdadeiros positivos (precisão) e quão relevante são esses valores (revocação ou sensibilidade). Ex: Suponha que o modelo para classificar vinhos bons e ruins identifica 7 vinhos como bons em um dataset contendo 9 vinhos bons e alguns ruins. Se 4 das identificações estão corretas, mas 3 são, na verdade, ruins, a precisão do modelo é 4/7 enquanto a sua revocação é 4/9.

Usaremos uma classificação onde vinhos de qualidade igual ou maior que 6 será tido como bom (1) e vinhos de qualidade menor que 6 serão tidos como ruins (0).

II. Análise

Exploração dos dados

Aqui temos um breve resumo de cada feature:

1 - fixed acidity: alguns ácidos envolvidos no vinho fazendo que ele não evapore rapidamente

2 - volatile acidity: quantidade de ácido acético no vinho que se caso for muito alto pode gerar um sabor de vinagre causando desconforto.

3 - citric acid: deixa o vinho com um “frescor” no sabor

4 - residual sugar: quantidade de açúcar no vinho após o processo de fermentação

5 - chlorides: a quantidade de sal no vinho

6 - free sulfur dioxide: previne o crescimento de bactérias no vinho e oxidação.

7 - total sulfur dioxide: em baixas quantidades faz com que SO2 (dióxido sulfurico) seja indetectável no vinho, já em altas doses pode ser percebido no gosto ou no cheiro

8 - density: densisade da água em relação a quantidade de alcóol e açúcar no vinho

9 - pH: quão ácido é um vinho numa escala de 0 (muito ácido) para 14 (muito básico). A maioria dos vinhos está em 3 e 4 nessa escala

10 - sulphates: contribue com SO2 para que tenha uma ação antioxidante e antimicrobiana no vinho

11 - alcohol: quantidade de alcool no vinho

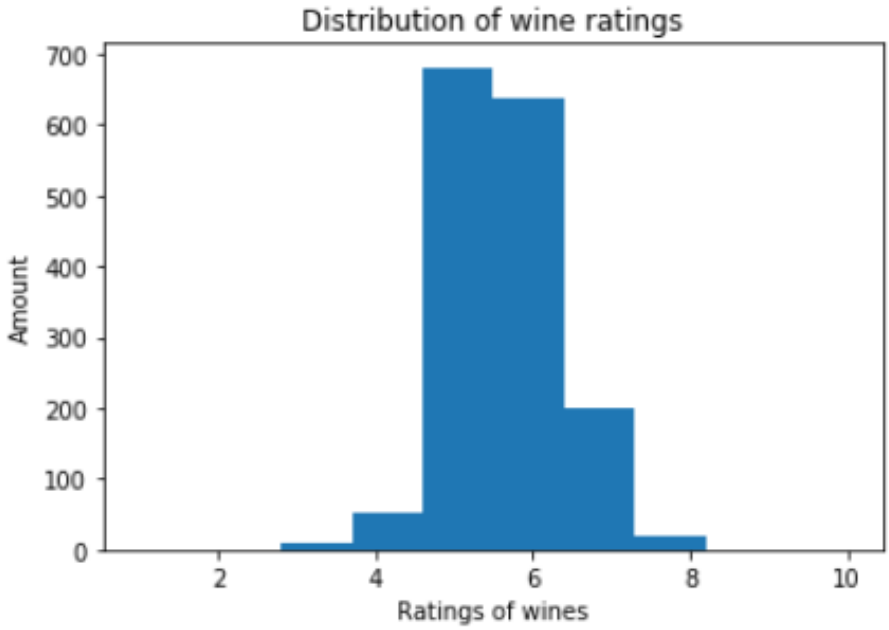
(Variável de saída) 12 – quality: qualidade do vinho numa escala de 0 a 10

Visualização exploratória

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Resumo das features no dataset

Aqui podemos ver alguns exemplos do nosso dataset para cada feature explicada anteriormente.



Distribuição dos vinhos (quantidade X qualidade)

Neste grafico podemos ver a variação de qualidade dos vinhos do nosso dataset. Temos um pico entre vinhos de qualidade 5 a 6 aproximadamente.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

Resumo de cada feature

Aqui temos a variação de cada feature como media, mediana, desvio padrão e etc.

Algoritmos e técnicas

Neste modelo utilizaremos a abordagem de classificação com aprendizagem supervisionada, onde vinhos de qualidade igual ou maior que 6 serão colocados como bons (1) e vinhos de qualidade menor que 6 serão colocados como ruins (0). Rodamos os testes com 3 classificadores diferentes: Gradient Boosting e Random Forest da família da árvore de decisão e SVM, Support Vector Machines método bastante eficaz de aprendizado supervisionado. Como classificadores podemos testar

essas 3 técnicas para gerar um modelo que receberá as informações de cada feature para cada exemplo e gerar uma saída de 0 ou 1, bom ou ruim.

Benchmark

Rodamos os testes para cada uma das 3 abordagens, dividimos o dataset em aproximadamente 70% para treino e 30% para teste, deste testamos o nossos classificadores em partes divididas, e o resultado foi o seguinte:

Classifer 1 - SVM

Training Set Size	Training Time	Prediction Time (test)	F1 Score (train)	F1 Score (test)
150	0.0055	0.0011	0.9452	0.5386
300	0.0056	0.0033	0.9315	0.5365
600	0.0199	0.0116	0.8917	0.6705

Classifer 2 - RandomForestClassifier

Training Set Size	Training Time	Prediction Time (test)	F1 Score (train)	F1 Score (test)
150	0.0703	0.0011	0.9863	0.7139
300	0.0239	0.0014	0.9347	0.7507
600	0.0437	0.0023	0.9208	0.7660

Classifer 3 - GradientBoostingClassifier

Training Set Size	Training Time	Prediction Time (test)	F1 Score (train)	F1 Score (test)
150	0.1360	0.0009	1.0000	0.7068
300	0.2504	0.0022	1.0000	0.7488
600	0.3435	0.0034	1.0000	0.7843

De acordo com essa tabela podemos comparar os resultados de treino e teste de cada classificador. Estou apenas levando em conta aqui o F1 Score de treino e teste, já que o tempo levado para treinar foi pouco. Antes de tudo queria deixar claro também que cada classificador teve uma pontuação razoável, mas Gradient Boosting tá tendo uma pontuação muito alta que pode caracterizar como overfitting, já o SVM está com a pontuação mais baixa entre todos. Por isso vou optar pelo Random Forest, que tem uma pontuação melhor que do SVM porém menor que a do Gradient Boosting que está um tanto exagerada.

III. Metodologia

Pré-processamento de dados

Nessa etapa de pré-processamento de dados devemos retirar algumas linhas consideradas como outliers utilizando o seguinte cálculo:

- se o valor de uma feature for maior que a média dessa feature menos duas vezes o desvio padrão; valor > média da feature – 2 * desvio padrão

- se o valor de uma feature for menor que a média dessa feature mais duas vezes o desvio padrão; valor < média da feature + 2 * desvio padrão

Nenhum outro tratamento precisa ser feito já que todos os dados são númericos.

Implementação

Como já dito antes, o classificador escolhido para esse modelo é o Random Forest pelos seguintes motivos:

- Random Forest Classifier cria um grupo de árvores de decisão randomicamente selecionadas, então utiliza os votos de cada árvore do grupo para decidir qual delas é a melhor
- Ele teve um score maior que o SVM tanto no treino quanto no teste
- Árvores de decisão podem ser aplicadas tanto em regressão quanto em classificação, sendo que em classificação o resultado é melhor e nosso modelo usará classificação onde cai muito bem
- Ele teve um score menor em treino comparado ao Gradient Boosting que no caso se mostrou em overfitting com score de 1.0 o que devemos evitar, e deixando a desejar na pontuação de teste que está muito próxima da pontuação do Random Forest
- O tempo de treinamento está ótimo
- Podemos utilizar alguns parametros para otimizar as pontuações de treino e teste

Refinamento

Não podemos testar todos os parametros possíveis, porém, temos aqui alguns deles que ajudaram no aumento da pontuação e precisão do modelo:

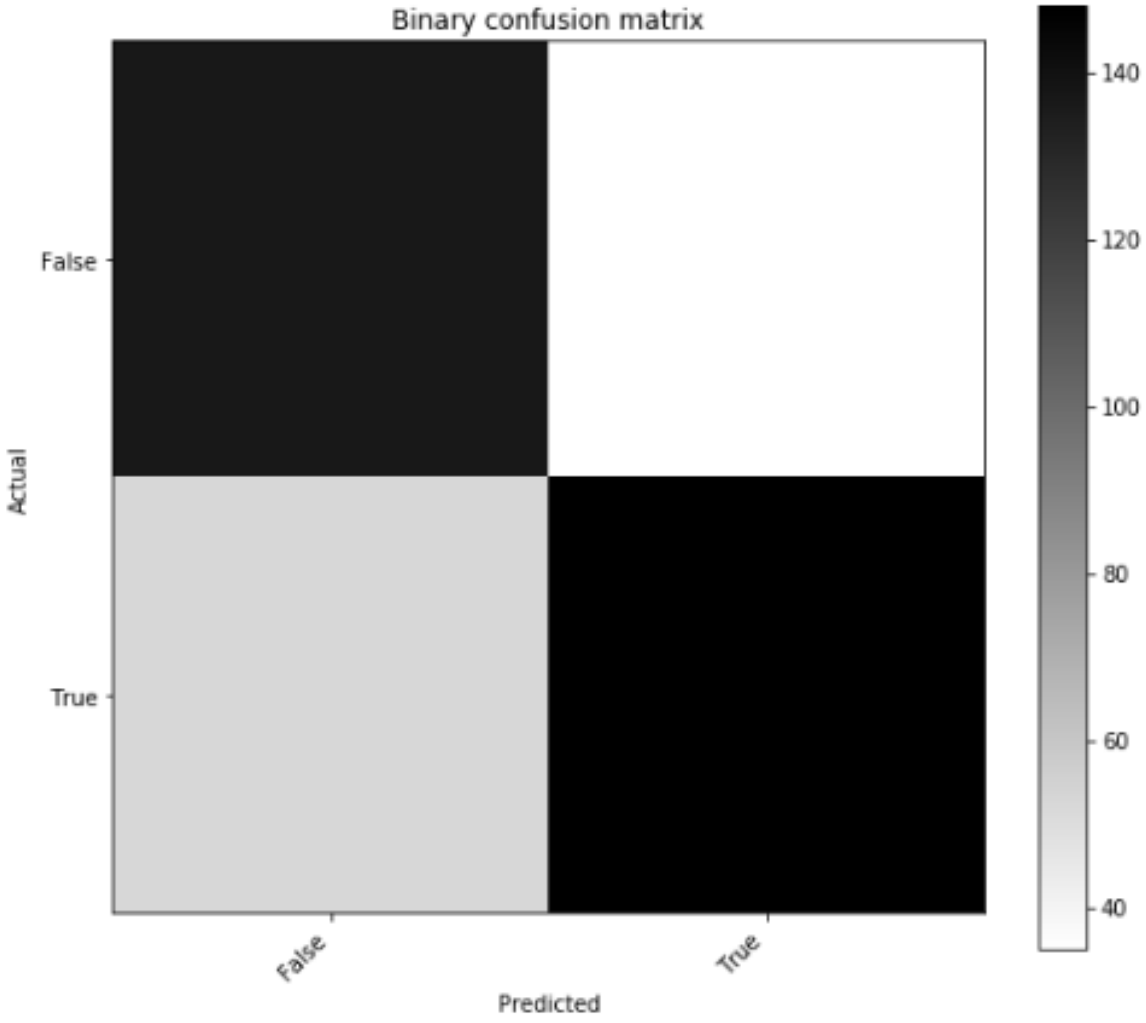
- n_estimators: o número de árvores utilizadas no modelo, neste caso utilizaremos 100
- min_samples_leaf: o número mínimo para cada nó (folha) da árvore
- max_depth: a profundidade máxima usada na árvore

O resultado desse modelo de Random Forest com os parametros foi:

- Treino feito em 0.0103 segundos com pontuação de 0.9393
- Teste feito em 0.0093 segundos com pontuação de 0.7708

IV. Resultados

Modelo de avaliação e validação



```
   Binary confusion matrix:
Predicted  False  True  __all__
Actual
False        137    35      172
True          53   148      201
__all__      190   183      373
```

Aqui podemos ver os valores apresentados a respeito de quão acertivo foi o nosso modelo. Temos valores muito bons onde apenas 35 de 172 vinhos ruins estão sendo classificados como bons e 53 de 201 vinhos bons sendo classificados como ruins. Não está ruim mas pode ser melhorado.

Justificativa

Benchmark:

- score de treino foi 0.9208.

- score de teste foi 0.7660.

No modelo otimizado:

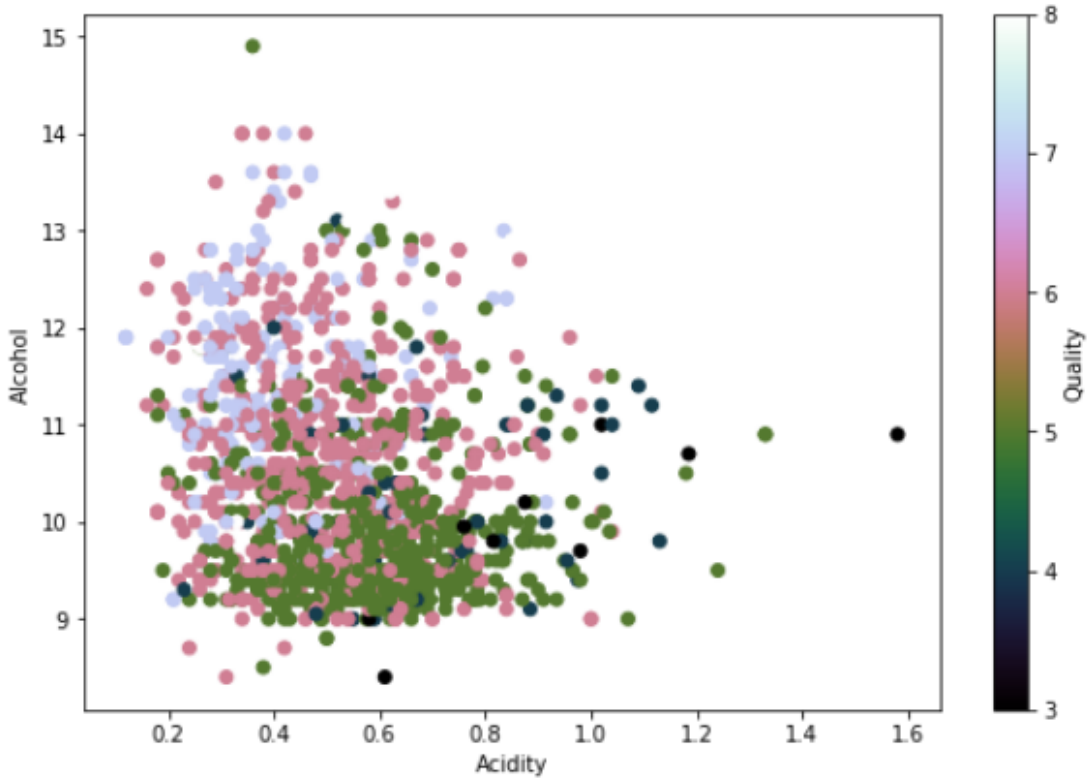
- score de treino foi 0.9393.

- score de teste foi 0.7708.

Conseguimos uma pequena melhora no score do nosso modelo otimizado comparado ao benchmark, talvez poderia ser feita uma melhoria no score de teste do nosso modelo utilizando alguns outros parametros que não conseguimos testar anteriormente. De toda forma, considero um bom modelo.

V. Conclusão

Foma livre de visualização



quantidade de álcool X acidez (volatile acidity) X qualidade

Fiz uma rápida análise a respeito das correlações das features com a qualidade do vinho, então cheguei ao resultado de que as features mais relevantes são a quantidade de álcool (influenciando positivamente) e acidez volátil (influenciando negativamente), ou seja: vinhos bons estão bem relacionados com grande quatidade de álcool e baixa acidez volátil.

Reflexão

O projeto foi um tanto quanto intrigante e desafiador para um iniciante. Fazer a abordagem de classificação pode ter sido uma boa forma de começar um modelo simples para resolver esse problema em questão.

O dataset tem um numero considerável de features a respeito de vinhos vermelhos cuja qualidade é pontuada de 0 a 10. Optei por utilizar apenas uma classificação simples onde vinhos de qualidade >= 6 seriam considerados bons e seriam colocados no grupo 1 e o restante seriam considerados ruins e seriam colocados no grupo 0.

Ao rodar alguns testes entre os 3 classificadores escolhidos (SVM, Random Forest e Gradient Boosting) optei pelo Random Forest, por questões de ter uma pontuação

melhor que a do SVM e ter uma inclinação para overfitting menor que a do Gradient Boosting.

No pré-processamento tiramos apenas alguns outliers, o dataset é bem conciso com apenas dados número sem a necessidade de manipular outro tipo de dados e sem dados faltando.

Refinamos o modelo utilizando um grid search com alguns parametro buscando uma melhoria na pontuação final para treino e teste.

A melhoria foi obtida com sucesso, não foi um número muito grande já que o nosso classificador já tinha uma pontuação aceitável.

Melhorias

- Pode ser feito outro tipo de modelo utilizando regressão, no caso da qualidade do vinho que é nossa variável de saída numa escala de 0 a 10
- Pode também ser utilizada a abordagem de redes neurais artificiais
- Não podemos garantir se algum desses exmplos pode ter alguma melhoria, mas seria interessante testar
- Talvez com mais um pouco de estudo eu possa futuramente fazer um modelo mais acertivo e robusto com essa mesma abordagem ou com as citadas acima