Mineração de Dados Complexos
Curso de Aperfeiçoamento
INF-0615 - Aprendizado de Máquina

Qualidade de Vinho
INF-0615 - Aprendizado de Máquina

Rafael Fernando Ribeiro
Thiago Gomes Marçal Pereira

Prof. Anderson Rocha

Agosto de 2018

Análise dos Dados

Para o desenvolvimento desse trabalho, partimos de um conjunto de dados de treinamento com :

- 3898 linhas de dados
- 12 colunas, sendo uma dessas colunas a classificação indicando se o vinho é bom ou não.

Numa análise inicial dos dados, obtivemos a seguinte divisão em quartis, bem como mínimos e máximos:

```
> summary(train_data)
 fixed.acidity    volatile.acidity  citric.acid     residual.sugar     chlorides       free.sulfur.dioxide
 Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600   Min.   :0.01200   Min.   :  1.00
 1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2400   1st Qu.: 1.800   1st Qu.:0.03800   1st Qu.: 17.00
 Median : 6.900   Median :0.2900   Median :0.3100   Median : 3.000   Median :0.04700   Median : 29.00
 Mean   : 7.195   Mean   :0.3386   Mean   :0.3169   Mean   : 5.418   Mean   :0.05584   Mean   : 30.64
 3rd Qu.: 7.600   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.000   3rd Qu.:0.06400   3rd Qu.: 41.00
 Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800   Max.   :0.61100   Max.   :146.50
 total.sulfur.dioxide    density             pH          sulphates         alcohol         quality
 Min.   :  6.00       Min.   :0.9871   Min.   :2.74   Min.   :0.2300   Min.   : 8.00   Min.   :0.0000
 1st Qu.: 76.25       1st Qu.:0.9924   1st Qu.:3.11   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:0.0000
 Median :118.00       Median :0.9949   Median :3.21   Median :0.5100   Median :10.30   Median :0.0000
 Mean   :115.33       Mean   :0.9947   Mean   :3.22   Mean   :0.5317   Mean   :10.47   Mean   :0.1978
 3rd Qu.:155.00       3rd Qu.:0.9970   3rd Qu.:3.32   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:0.0000
 Max.   :366.50       Max.   :1.0390   Max.   :4.01   Max.   :2.0000   Max.   :14.90   Max.   :1.0000
```

A maiorida dos dados parece apresentar Outliers, com máximos fora das faixas médias, principalmente, *fixed.acidity, volatile.avidity, citric.acid, residual.sugar, chlorides, free.sulfur.dioxide, total.sulfur.dioxide and sulphates.* Para melhor análise, num segundo momento, todos os dados serão considerados na faixa média +/- 2*desvio padrão.

Pudemos também perceber que a quantidade de dados contém uma maior quantidade de vinhos considerados de baixa qualidade, o que prejudica a análise dos dados, como veremos mais à frente. Portanto, será também feita uma análise com os dados balanceados, para isso, serão utilizados 2 métodos:

- redução dos dados para balanceamento
- criação de novos dados com SMOTE.

Treinamento

O treinamento dos dados se deu utilizando regressão logística. Para comparação dos dados, utilizou-se diversos valores de lambda, variando de de 10^{-5} até 10. Inicialmente os dados foram treinados conforme obtidos. Num momento seguinte, foram balanceados utilizando os dois métodos mencionados.

Após esses resultados, foram removidos os Outliers, e o mesmo processo se repetiu, utilizando-se os balanceamentos.

Definindo-se então a melhor combinação de informação dos dados, balanceados ou não, com ou sem outliers e valor de lambda, partimos para uma tentativa de melhoria do treinamento fazendo combinação de features existentes.

Cabe ressaltar, que a remoção de outliers foi realizada apenas no conjunto de treinamento.

Para cálculo da acurácia, utilizamos matriz de confusão, e o cálculo a partir dos dados normalizados por classe. Os resultados obtidos foram:

```
> logistic.train(train_data, val_data, test_data)
  lambda  accTrain    accVal   accTest
1  0e+00 0.6096081 0.5693612 0.5675893
2  1e-05 0.6097680 0.5693612 0.5675893
3  1e-04 0.6099279 0.5693612 0.5675893
4  1e-03 0.6066764 0.5681650 0.5667056
5  1e-02 0.5934750 0.5608717 0.5554274
6  1e-01 0.5264646 0.5086057 0.5119048
7  0e+00 0.6096081 0.5693612 0.5675893
8  1e+00 0.5000000 0.5000000 0.5000000
9  1e+01 0.5000000 0.5000000 0.5000000
```

```
> logistic.train(balanced_train, val_data, test_data)
  lambda  accTrain    accVal   accTest
1  0e+00 0.7230869 0.7105573 0.7085543
2  1e-05 0.7230869 0.7111554 0.7085543
3  1e-04 0.7230869 0.7111554 0.7091808
4  1e-03 0.7243839 0.7093611 0.7108034
5  1e-02 0.7354086 0.7118697 0.7098074
6  1e-01 0.7425422 0.7105573 0.7029152
7  0e+00 0.7230869 0.7105573 0.7085543
8  1e+00 0.7269780 0.6949735 0.6917019
9  1e+01 0.7185473 0.6777117 0.6844726
```

```
> logistic.train(smoted_data, val_data, test_data)
  lambda  accTrain    accVal   accTest
1  0e+00 0.7534371 0.7103248 0.7151894
2  1e-05 0.7535668 0.7103248 0.7151894
3  1e-04 0.7540856 0.7103248 0.7145628
4  1e-03 0.7559014 0.7097267 0.7133097
5  1e-02 0.7553826 0.7114047 0.7044254
6  1e-01 0.7522698 0.7129497 0.7044254
7  0e+00 0.7534371 0.7103248 0.7151894
8  1e+00 0.7320363 0.6923318 0.6892281
9  1e+01 0.7252918 0.6805858 0.6826254
```

E para os dados sem Outliers:

```
> logistic.train(train_data, val_data, test_data)
  lambda  accTrain    accVal   accTest
1  0e+00 0.6031053 0.5666033 0.5716058
2  1e-05 0.6031053 0.5666033 0.5716058
3  1e-04 0.6031053 0.5666033 0.5716058
4  1e-03 0.6045696 0.5670851 0.5716058
5  1e-02 0.5917551 0.5627823 0.5670751
6  1e-01 0.5181506 0.5187732 0.5228135
7  0e+00 0.6031053 0.5666033 0.5716058
8  1e+00 0.5000000 0.5000000 0.5000000
9  1e+01 0.5000000 0.5000000 0.5000000
```

```
> logistic.train(balanced_train, val_data, test_data)
  lambda  accTrain    accVal   accTest
1  0e+00 0.7396226 0.6973168 0.6914597
2  1e-05 0.7396226 0.6973168 0.6914597
3  1e-04 0.7405660 0.6973168 0.6920863
4  1e-03 0.7405660 0.6963531 0.6890982
5  1e-02 0.7424528 0.6912028 0.6934842
6  1e-01 0.7452830 0.6817665 0.6792978
7  0e+00 0.7396226 0.6973168 0.6914597
8  1e+00 0.7254717 0.6708679 0.6669438
9  1e+01 0.7141509 0.6642722 0.6584289
```

```
> logistic.train(smoted_data, val_data, test_data)
  lambda  accTrain    accVal   accTest
1  0e+00 0.7250943 0.6928808 0.6886165
2  1e-05 0.7250943 0.6928808 0.6886165
3  1e-04 0.7250943 0.6906048 0.6898696
4  1e-03 0.7275472 0.6912028 0.6908656
5  1e-02 0.7303774 0.6943263 0.6958457
6  1e-01 0.7316981 0.6882460 0.6872660
7  0e+00 0.7250943 0.6928808 0.6886165
8  1e+00 0.7152830 0.6699210 0.6661724
9  1e+01 0.7128302 0.6606837 0.6549267
```

Como podemos perceber, a melhor acurácia nos dados de treinamento, foi obtida para os dados sem remoção de Outliers, realizado o balanceamento com SMOTE, e com Lambda 10^{-3}.

Utilizaremos então esses dados para tentar uma melhora no treinamento.

Rodando uma correlação entre os dados, após balanceamento, percebemos que os valores com menor correlação são:

- chlorides x citric.acid
- chlorides x pH
- density x total.sulfur.dioxide
- density x free.sulfur.dioxide

Então partiremos desses dados para combinação de features.

As várias tentativas realizadas de combinação de features, bem como aumento do grau do polinômio, se mostraram infrutíferas, resultando em uma diminuição da acurácia, em mais de 15% no conjunto de treinamento e de mais de 20% nos conjuntos de validação e treinamento.

Conclusão

Assim como no desenvolvimento do primeiro trabalho, foi interessante utilizar métodos de treinamento de modelos para definição de classificação e decisão.

Após o desenvolvimento desse trabalho no entanto, fica mais claro que carecemos ainda de mais experiência para determinação de como melhorar o modelo para obtermos maior acurácia nos resultados obtidos. E que como muitas vezes explicado em sala de aula, não há uma receita ou uma opção que sempre servirá. Cada caso deve ser analisado, e podemos partir de princípios, mas sempre nos mantendo atentos a opções que possam ser de maior valia.

Um bom exemplo disso é que a remoção dos possíveis Outliers do conjunto de treinamento não resultou em uma melhora do modelo, pelo contrário, representou uma redução de alguns pontos percentuais na acurácia.