RESPOSTAS ÀS PERGUNTAS DO DESAFIO:

A análise exploratória completa dos dados se encontra no NOTEBOOK 2; conjuntamente com uma última sessão de treinamento de SVM’s com GridSearch com menor parcimoniosidade com relação à identificação e eliminação de outliers do banco de dados.

No primeiro notebook (NOTEBOOK 1), temos diversas sessões de treinamento de SVM’s com GridSearch e também uma sessão de treinamento de Random Forests com Grid Search. Nesta última sessão de treinamento, ranqueou-se as features utilizadas no treinamento de classificadores Random Forest com o método feature\_importances\_. A seguir expõe-se o ranqueamento (em ordem decrescente) da importância das features utilizadas obtido:

1) chlorides e alcohol

2) total sulfur dioxide

3) fixed acidity e volatile acidity

4) pH

5) citric acid e residual sugar

6) sulphates

7) type

AS diversas sessões de treinamento com GridSearch de classificadores SVM, no NOTEBOOK 1, variaram os possíveis valores que os hiper parâmetros poderiam assumir, o kernel utilizado e a função de scoring a ser utilizada nos scores de validação cruzada (esses scores, por sua vez, serão utilizados para a escolha do melhor modelo, em cada sessão de treinamento com Grid Search.

O NOTEBOOK 1 também inclui uma sessão de treinamentos de SVM’s com GridSearch e reamostragem Naive no conjunto de treinamento visando-se eliminar o desbalanceamento de classes no conjunto de dados original.

As funções de score utilizadas nas validações cruzadas dos treinamentos dos modelos em Grid Search, foram a Acurácia e o F1-Score. Os motivos pelos quais se utilizou essas duas métricas no scoring de validação cruzada serão explicados ao longo desse relatório.

Como já mencionado anteriormente, o NOTEBOOK 2 apresenta uma última sessão de treinamento de classificadores SVM, só que dessa vez, utilizando um dataset inicialmente mais reduzido pela aplicação de critérios menos parcimoniosos na identificação e eliminação de outliers. A seguir, esse dataset foi amplido também com o uso de técnicas de reamostragem (up sampling) visando balancear as classes da variável resposta (inicialmente desbalanceadas) com o algoritmo SMOTE-NC.

1. A definição da estratégia de modelagem, dependeu em parte das particularidades dos dados do desafio e da intenção inicial de se utilizar Support Vector Machines (SVM’s) para se construir um classificador preditivo da qualidade do vinho com base nas features disponíveis e em parte das melhores práticas de Data Science relativas a todo o workflow desenvolvido (tanto no que diz respeito às etapas de pré-processamento como às etapas de treinamento com GridSearch e escolha do melhor modelo baseado em scores de Validação Cruzada).

Primeiramente, optou-se por SVM’s por serem estes modelos capazes de fornecer classificadores preditivos de muito bom desempenho na literatura e em aplicações corporativas (para uma grande gama de métricas de avaliação de performance preditiva).

Uma vez tendo se feito essa escolha, foi necessário se proceder a uma série de atividades de *data wrangling* e *feature engineering* para que se pudesse assegurar um

adequado treinamento de diferentes SVM’s com diferentes combinações de hiperparÂmetros .

Nesse sentido, primeiramente se checou se a base dados apresentava missing values, o que não se constatou.

A seguir se procedeu à eliminação de outliers nas features disponíveis. Como se trata de um banco de dados relativamente pequeno ( e portanto ser possível que os dados presentes não cubram toda a variabilidade presente, na população, nas variáveis consideradas), se optou por uma abordagem mais parcimoniosa na identificação de possíveis outliers; considerando-se outliers somente observações das features que se encontrassem acima ou abaixo de quatro intervalos interquartis, de acordo com a distribuição empírica de cada feature (contínua).

A seguir se procedeu à investigação de padrões de correlação entre cada par de feature contínua, se plotando uma matriz de scatter plots para cada par dessas features; assim como uma matriz de coeficientes de correlação de Pearson entre pares de features.

Com a matriz de plots se buscou identificar padrões persistentes de correlação (linear ou não linear) entre as features, identificando-se alguns padrões de correlação já um pouco evidentes (embora não extremamente acentuados entre as features.

Com a matriz de correlações de Pearson, se encontro um par de features com valor mais importante. Isso ocorreu entre as features total sulfur dioxide e free sulfur dioxide, com valor de correlação acima de 0,7.

Deste modo, optou-se por eliminar a feature free súlfur dioxide, pois:

1. O valor da correlação não é desprezível,
2. Features com altos valores nesse tipo de correlação implica em problemas de estimação de SVM’s e geralmente implica em modelos mais complexos do que o necessário ou demandam a utilização de um algoritmo de feature selection para o SVM.

Também se eliminou a feature density, que possuía correlação, em módulo, maior que 0.7, com a feature alcohol. Adicionalmente, a feature density possuía uma distribuição atípica ao longo de todos os labels da variável resposta, com muitos valore muito baixos e muitos valores muito altos, e nenhum valor intermediário; isso implicaria que a adoção de um critério eliminador de outliers iria eliminar uma grande quantidade de observações dessa feature.

Com relação ao treinamento propriamente dito das máquinas de aprendizado:

Se procedeu ao treinamento de classificadores tanto baseados em SVM como em Random Forests, em alguns casos, usando-se como score de validação cruzada a métrica F1, com micro average; e em outros a acurácia..

A escolha da métricaF1 como score de cross-validação se deveu ao fato de que, ao se treinar um classificador preditivo de qualidades do vinho, em geral se deseja que esse classificador tenha as seguintes propriedades:

1. Que, ao fazer uma predição sobre a qualidade de um vinho específico, se gostaria que se estivesse muito certo sore esta qualidade (ou seja, ou seja, se gostaria de uma Precisão elevada)
2. Se gostaria de minimizar ao máximo casos em que o classificador preveja um vinho como não tendo uma má propriedade (por exemplo, baixa qualidade) quando na verdade ele tem baixa qualidade; assim se fôssemos com algum critério, dividirmos os diferentes valores (labels) de qualidade de vinho, em duas grandes categorias (boa qualidade e má qualidade) gostaríamos também de minimizar os casos em que o classificador diz que um vinho tem boa qualidade quando na verdade tem má (isto é, gostaríamos de minimizar os casos de Falso Negativos), o que equivale a maximizar o Recall.
3. Como se sabe ao se tentar maximizar um dos objetivos (Precision ou Recall) tende-se a piorar o outro, pois há um tradeoff entre os dois, recomenda-se uma métrica que combine as duas métricas consideradas, como o F1-score, que é a média harmônica de Precision e Recall.

A escolha da Acurácia se deveu ao fato ser uma métrica comumente adotada em problemas e classificação com classes balanceadas (o que foi obtido com resampling).

No treinamento dos classificadores foram utilizados GridSearch usando-se 4 folds de validação cruzada. Optou-se por este número de folds para que o tempo de computação envolvido no treinamento dos modelos não fosse muito alto e houvesse tempo suficiente para se analisar os resultados encontrados e se efetuar esse relatório dentro do tempo disponível.

Somente em uma das sessões de treinamento de SVM’s se utilizou como score de cross validação a Acurácia, para fins de comparação de resultados com os demais classificadores treinados (que utilizaram como score de validação cruzada, a métrica F1-score).

Em quase todos os modelos treinados, utilizou-se o algoritmo SMOTE-NC de resampling para se rebalancear as classes da variável resposta.

Somente em um caso se treinou classificadores SVM com uma estratégia de resampling simples, ou Naive, se replicando os casos com classes minoritárias, para fins de comparação com os resultados obtidos com outras estratégias mais sofisticadas de resampling.

1. A função de custo utilizada nos classificadores SVM é a utilizada por default , para este modelo, na biblioteca sklearn (versão 0.20.3): a função de custo é baseada na função de perda de hinge (mais particularmente hinge ao quadrado).

A seguir se apresentará a função de perda e a correspondente função de custo usada pelos classificadores SVM treinados (correspondentes a classe SVC da biblioteca sklearn):

Função de perda de hinge:





Essa função de perda é apropriada para o objetivo da maximização da margem entre os dados e o hiperplano. Com esta função, o custo é zero se o valor previsto pelo modelo e o valor verdadeiro (no caso do problema analisado, a classe verdadeira da variável resposta) são do mesmo sinal; caso contrário, o valor de perda 1 – y \* f(x) é computado.

Com base nessa função de perda, a função de custo é computada como abaixo, com uma observação adicional de que nela há também um termo de regularização L2 para balancear a maximização da margem e perda.

A função de custo associada à função de Hinge dada acima é:



Referências:

<https://scikit-learn.org/stable/auto_examples/svm/plot_svm_scale_c.html>

Função de custo utilizada nos classificadores Random Forest na biblioteca sklearn:

1. O critério utilizado para a seleção do modelo final, foi o seguinte: primeiramente selecionou-se o melhor modelo em cada sessão de treinamento de GridSearch, com base na função de scoring (na validação cruzada) utilizada em cada sessão. Depois se selecionou o melhor modelo, dentre estes melhores, com base na ponderação entre o F1-score micro average no conjunto de treinamento e de teste. O modelo final selecionado foi o que apresentou melhor desempenho no conjunto de teste e ao mesmo tempo não mostrou indícios de overfitting no conjunto de treinamento. O modelo final foi o modelo 5, como descrito mais abaixo no relatório.
2. Os modelos treinados selecionados (isto é, o melhor modelo de cada sessão de treinamentoforam validados com base na consideração conjunta do score F1-micro average no conjunto de treinamento e no mesmo score no conjunto de teste. O modelo final escolhido foi o melhor dentre os melhores de cada sessão de treinamento GridSearch, com base nos mesmo critérios.
3. As evidências de que o **modelo final selecionado (modelo 5)** é um bom modelo estão presentes nos relatórios de classificação, tanto no conjunto de treinamento como no de teste. No de treinamento o modelo obteve precisão de 0.81, recall de 0.71 e F1-score (média harmônica dos dois scores anteriores) de 0.71; não apresentando indícios de overfitting.

No conjunto de teste, o modelo apresentou o valor de 0.93 para os três scores, sendo o modelo que melhor performou nesse conjunto.

A seguir apresentamos as características do melhor selecionado em cada sessão de treinamento GrisSearch e seus respectivos relatórios de classificação.

Modelo 5 (melhor modelo da 5ª sessão de treinamento com GridSearch):

**MODELO FINAL ESCOLHIDO!!!**

# Treinando classificador usando como CV-score a Acurácia e Kernel RBF no conjunto de treinamento

# obtido com resampling Naive

best\_params\_: {'C': 100, 'gamma': 1, 'kernel': 'rbf'}

Best estimator:

SVC(C=100, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=1, kernel='rbf',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

Teste

precision recall f1-score support

3 1.00 1.00 1.00 665

4 0.99 1.00 0.99 656

5 0.83 0.86 0.84 664

6 0.82 0.73 0.77 686

7 0.90 0.95 0.93 686

8 0.99 1.00 0.99 692

9 1.00 1.00 1.00 689

micro avg 0.93 0.93 0.93 4738

macro avg 0.93 0.93 0.93 4738

weighted avg 0.93 0.93 0.93 4738

train:

precision recall f1-score support

3 1.00 0.41 0.59 1914

4 0.97 0.60 0.74 1914

5 0.68 0.83 0.75 1914

6 0.44 0.93 0.59 1914

7 0.72 0.81 0.77 1914

8 0.86 0.71 0.78 1914

9 1.00 0.66 0.80 1914

micro avg 0.71 0.71 0.71 13398

macro avg 0.81 0.71 0.72 13398

weighted avg 0.81 0.71 0.72 13398

Modelo 4: (melhor modelo da 4ª sessão de treinamento com GridSearch):

SVM, com kernel Linear, e conjunto de treinamento com classes balanceadas

# por meio de reamostragem (superamostragem, ou up sampling das classes minoritárias) com o algoritmo SMOTE-NC

# E uso de Grid Search, e usando como scoring de cross validação, a métrica F1-score, com estratégia de micro average

# Cross Validação com 4-fold

best\_params\_: {'C': 7, 'kernel': 'linear'}

Best estimator:

SVC(C=7, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma='auto\_deprecated',

kernel='linear', max\_iter=-1, probability=False, random\_state=None,

shrinking=True, tol=0.001, verbose=False)

Teste

precision recall f1-score support

3 0.00 0.11 0.01 9

4 0.05 0.16 0.08 62

5 0.54 0.27 0.36 616

6 0.61 0.20 0.30 793

7 0.31 0.36 0.33 318

8 0.09 0.53 0.15 59

9 0.03 1.00 0.07 1

micro avg 0.26 0.26 0.26 1858

macro avg 0.23 0.38 0.19 1858

weighted avg 0.50 0.26 0.31 1858

train:

precision recall f1-score support

3 0.55 0.71 0.62 1914

4 0.53 0.43 0.47 1914

5 0.39 0.42 0.41 1914

6 0.32 0.22 0.26 1914

7 0.41 0.38 0.39 1914

8 0.50 0.62 0.55 1914

9 0.99 1.00 0.99 1914

micro avg 0.54 0.54 0.54 13398

macro avg 0.53 0.54 0.53 13398

weighted avg 0.53 0.54 0.53 13398

Modelo 3: (melhor modelo da 3ª sessão de treinamento com GridSearch):

SVM, com kernel Linear, e conjunto de treinamento com classes balanceadas

# por meio de reamostragem (superamostragem, ou up sampling das classes minoritárias) com o algoritmo SMOTE-NC

# E uso de Grid Search, e usando como scoring de cross validação, a métrica F1-score, com estratégia de micro average

# Cross Validação com 4-fold

best\_params\_: {'bootstrap': False, 'criterion': 'gini', 'n\_estimators': 1000}

Best estimator:

RandomForestClassifier(bootstrap=False, class\_weight=None, criterion='gini',

max\_depth=None, max\_features='auto', max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=1000, n\_jobs=None,

oob\_score=False, random\_state=0, verbose=0, warm\_start=False)

Teste

precision recall f1-score support

3 0.00 0.00 0.00 9

4 0.14 0.31 0.20 62

5 0.69 0.58 0.63 616

6 0.61 0.54 0.57 793

7 0.45 0.60 0.52 318

8 0.27 0.32 0.29 59

9 0.00 0.00 0.00 1

micro avg 0.55 0.55 0.55 1858

macro avg 0.31 0.34 0.32 1858

weighted avg 0.58 0.55 0.56 1858

train:

precision recall f1-score support

3 1.00 1.00 1.00 1914

4 1.00 1.00 1.00 1914

5 1.00 1.00 1.00 1914

6 1.00 1.00 1.00 1914

7 1.00 1.00 1.00 1914

8 1.00 1.00 1.00 1914

9 1.00 1.00 1.00 1914

micro avg 1.00 1.00 1.00 13398

macro avg 1.00 1.00 1.00 13398

weighted avg 1.00 1.00 1.00 13398

Modelo 2: (melhor modelo da 2ª sessão de treinamento com GridSearch):

SVM, com kernel RBF, e conjunto de treinamento com classes balanceadas

# por meio de reamostragem (superamostragem, ou up sampling das classes minoritárias) com o algoritmo SMOTE-NC

# E uso de Grid Search, e usando como scoring de cross validação, a métrica Acurácia, com estratégia de micro average

# Cross Validação com 4-fold

best\_params\_: {'C': 8, 'gamma': 1, 'kernel': 'rbf'}

Best estimator:

SVC(C=8, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=1, kernel='rbf',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

Teste

precision recall f1-score support

3 0.00 0.00 0.00 9

4 0.16 0.21 0.18 62

5 0.67 0.58 0.62 616

6 0.59 0.64 0.62 793

7 0.55 0.55 0.55 318

8 0.35 0.34 0.34 59

9 0.00 0.00 0.00 1

micro avg 0.58 0.58 0.58 1858

macro avg 0.33 0.33 0.33 1858

weighted avg 0.58 0.58 0.58 1858

train:

precision recall f1-score support

3 1.00 1.00 1.00 1914

4 1.00 1.00 1.00 1914

5 1.00 0.99 0.99 1914

6 0.99 0.99 0.99 1914

7 0.99 1.00 0.99 1914

8 1.00 1.00 1.00 1914

9 1.00 1.00 1.00 1914

micro avg 1.00 1.00 1.00 13398

macro avg 1.00 1.00 1.00 13398

weighted avg 1.00 1.00 1.00 13398

Modelo 1: (melhor modelo da 1ª sessão de treinamento com GridSearch):

SVM, com kernel RBF, e conjunto de treinamento com classes balanceadas

# por meio de reamostragem (superamostragem, ou up sampling das classes minoritárias) com o algoritmo SMOTE-NC

# E uso de Grid Search, e usando como scoring de cross validação, a métrica Acurácia, com estratégia de micro average

# Cross Validação com 4-fold

best\_params\_: {'C': 8, 'gamma': 1, 'kernel': 'rbf'}

Best estimator:

SVC(C=8, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=1, kernel='rbf',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

Teste

precision recall f1-score support

3 0.00 0.00 0.00 9

4 0.16 0.21 0.18 62

5 0.67 0.58 0.62 616

6 0.59 0.64 0.62 793

7 0.55 0.55 0.55 318

8 0.35 0.34 0.34 59

9 0.00 0.00 0.00 1

micro avg 0.58 0.58 0.58 1858

macro avg 0.33 0.33 0.33 1858

weighted avg 0.58 0.58 0.58 1858

train:

precision recall f1-score support

3 1.00 1.00 1.00 1914

4 1.00 1.00 1.00 1914

5 1.00 0.99 0.99 1914

6 0.99 0.99 0.99 1914

7 0.99 1.00 0.99 1914

8 1.00 1.00 1.00 1914

9 1.00 1.00 1.00 1914

micro avg 1.00 1.00 1.00 13398

macro avg 1.00 1.00 1.00 13398

weighted avg 1.00 1.00 1.00 13398

Modelo 6: (melhor modelo da 6ª e última sessão de treinamento com GridSearch, no NOTEBOOK 2):

SVM, com kernel RBF, e conjunto de treinamento com classes balanceadas

# por meio de reamostragem (superamostragem, ou up sampling das classes minoritárias) com o algoritmo SMOTE-NC

# E uso de Grid Search, e usando como scoring de cross validação, a métrica F1, com estratégia de micro average

# Cross Validação com 4-fold. Treinamento efetuado com um critério menos parcimonioso na eliminação de outliers.

best\_params\_: {'C': 6, 'gamma': 1, 'kernel': 'rbf'}

Best estimator:

SVC(C=6, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=1, kernel='rbf',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

Teste

precision recall f1-score support

3 1.00 1.00 1.00 1806

4 1.00 1.00 1.00 1806

5 0.99 1.00 0.99 1806

6 0.99 0.99 0.99 1806

7 1.00 1.00 1.00 1806

8 1.00 1.00 1.00 1806

9 1.00 1.00 1.00 1806

micro avg 1.00 1.00 1.00 12642

macro avg 1.00 1.00 1.00 12642

weighted avg 1.00 1.00 1.00 12642

train:

precision recall f1-score support

3 1.00 1.00 1.00 1914

4 1.00 1.00 1.00 1914

5 1.00 0.99 0.99 1914

6 0.99 0.99 0.99 1914

7 0.99 1.00 0.99 1914

8 1.00 1.00 1.00 1914

9 1.00 1.00 1.00 1914

micro avg 1.00 1.00 1.00 13398

macro avg 1.00 1.00 1.00 13398

weighted avg 1.00 1.00 1.00 13398

CONSIDERAÇÕES FINAIS:

Os desenvolvimentos desse trabalho visaram construir classificadores preditivos da qualidade do vinho a partir da features fornecidas, utilizando estratégias de up sampling das classes minoritárias da variável resposta (inclusive utilizando o algoritmo SMOTE NC que é um dos melhores de upsampling no momento) .

Mesmo com essas estratégias e implementação de diversos esquemas de treimento Grid Search, alguns dos classificadores encontrados claramente se super ajustaram ao conjunto de treinamento (isto é, apresentaram sintomas de overffiting, com superajustamento ao conjunto de treinamento e pouca generalização no conjunto de teste).

Acredita-se que se possa melhorar ainda mais o desempenho dos classificadores por meio da combinação das seguintes estratégias:

1. Atribuição de pesos maiores Às clases minoritárias
2. Consideração de gamas maiores de valores de parâmetros/hiperparÂmetros nos Grid Searches,
3. Utilização de outras estratégias de re-amostragem, tanto de up sampling das classes minoritárias, como de subsampling das classes majoritárias assim como uma combinação das duas estratégias.

Contatos: [alfredorp33@yahoo.com.br](mailto:alfredorp33@yahoo.com.br)

Cel: +55 21 99987-2169

+55 11 99987-2154