Engenheiro de Aprendizado de Máquina Nanodegree

Projeto de conclusão de machine learning

Projeto: Prevendo a existência de doenças cardíacas (Heart Disease UCI)

Carlos Alberto dos Santos
12 de julho de 2019

I. Definição

Visão Geral do Projeto

Doença cardíaca é um termo geral para informar as várias condições médicas crônicas ou aquelas que afetam um ou mais componentes do coração. Existem diversas formas de avaliar e diagnosticar a existência dessa doença (exames, sintomas e outros). Uma análise, realizada no ano de 2015, estimou que 17,7 milhões de pessoas morreram por doenças cardiovasculares, representando 31% de todas as mortes em nível global. Esses números indicam a grande relevância do desenvolvimento de estudos em relação ao tema.

Estudos apontam que o Machine Learning tornou-se disseminado e indispensável para a resolução de problemas complexos nos diversos campos da ciência, sendo que na área médica sua utilização irá transformar a prática.

Esse projeto busca verificar a possibilidade da existência de doenças cardíacas em pessoas com a utilização de técnicas de Análise de Dados e algoritmos de Machine Learning. Para tanto, será utilizado um conjunto de dados (Heart Disease UCI) oriundo de uma pesquisa que envolveu 5 Instituições Médicas, de mais de um país, que obteve informações de pacientes com objetivo de identificar a existência de doença cardíaca. Esse conjunto de dados é um subconjunto com 14 atributos (originalmente eram 76 atributos) e 303 entradas que fizeram parte da pesquisa da Cleveland Clinic Foundation.

Declaração do problema

Trata-se de um problema que envolve a verificação da possibilidade da presença de doenças cardíacas em pessoas com base em informações de procedimentos quando comparadas às extraídas de um dataset no qual constam dados dos resultados de exames, testes e outros procedimentos anteriormente realizados por diversas pessoas por ocasião da pesquisa da Cleveland Clinic Foundation.

O objetivo desse projeto é aplicar algoritmos de Machine Learning, em especial de aprendizagem supervisionada, para treinar e testar modelos na busca dos melhores resultados na identificação da existência de doença cardíaca, sendo que a solução pode ser obtida por meio da variável alvo (target), tendo como resultados possíveis : 0 para negativo e 1 para positivo. Também será

realizada uma análise nas características dos dados para tentar encontrar quaisquer outras tendências que possam ajudar a prever certos eventos cardíacos, em especial, por meio da análise de correlação dos atributos.

Métricas

No projeto são utilizadas as seguintes métricas de avaliação para algoritmos de Machine Learning:

- Acurácia: divisão entre todos os acertos pelo total;
- Precisão: relação entre as previsões positivas realizadas corretamente e todas as previsões positivas (incluindo as falsas);
- Recall: relação entre as previsões positivas realizadas corretamente e todas as previsões que realmente são positivas (True Positives e False Negatives);
- F1-score: forma de resultado que utiliza as métricas Precision e Recall juntas.

Para análise do resultado final do projeto, devido às características do estudo, será utilizada, num primeiro momento a Acurácia que é o indicador mais simples de se calcular e não é afetado negativamente quando há classes balanceadas. Contudo, para ratificar a escolha do modelo também serão utilizadas as métricas Precisão, Recall e F1-score.

A precisão informa que as pessoas indicadas com resultado positivo teriam grande probabilidade de terem, efetivamente, a doença, buscando valorizar o menor resultado de falsos positivos.

O recall identificaria pessoas possivelmente com resultado positivo para uma primeira análise em razão da divisão das classes, mas necessitaria de um trabalho posterior de investigação. Essa métrica é útil quando se quer minimizar os falsos negativos. No caso em questão, pode haver um dano muito maior em não identificar a doença, do que identificá-la em pacientes saudáveis.

A métrica F1 Score é uma maneira de visualizar Precisão e Recall juntas e, para parte dos problemas, ela é considerada uma métrica melhor que a Acurácia.

II. Análise

Exploração de Dados

O conjunto Heart Disease UCI utilizado nesse projeto possui 14 atributos e 303 entradas. Os atributos possuem dados categóricos como “sex”, “fasting blood sugar”, “exercise induced angina” e “target”, bem como possui dados quantitativos como “age” (discretos, neste caso por se tratar de idade em anos) e contínuos como “chol” (serum cholestoral in mg/dl).

Lista dos atributos:

1. age (idade em anos);
2. sex (1 para masculino e 0 para feminino);
3. chest pain type (tipo de dor no peito, dividada em 4 valores);
4. resting blood pressure (pressão arterial em repouso (em mmHg na admissão ao hospital));
5. serum cholestoral in mg/dl (colesterol sérico);
6. fasting blood sugar > 120 mg/dl (açucar no sangue em jejum ((1 = verdadeiro; 0 = falso));
7. resting electrocardiographic results (resultados eletrocardiográficos em repouso(valores 0,1,2));
8. maximum heart rate achieved (frequência cardíaca máxima alcançada);

9. exercise induced angina (angina induzida por exercício (1 = sim; 0 = não));
10. oldpeak = ST depression induced by exercise relative to rest (depressão do segmento ST induzida pelo exercício em relação ao repouso);
11. the slope of the peak exercise ST segment (a inclinação do segmento ST de pico do exercício);
12. number of major vessels (0-3) colored by flourosopy;
13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect.;
14. target (VARIAVEL ALVO) = sendo 1 para possuí doença e 0 para não possuí doença.

Explorando os dados foi possível constatar que:

- as classes da variável alvo encontram-se balanceadas, com uma pequena diferença (percentual de indivíduos com resultado positivo de 54.46% e com resultado negativo de 45.54%);
- não foram localizados valores nulos ou ausentes;
- todos os dados são numéricos;
- existem outliers nos atributos "trestbps", "chol" e "thalach";
- localizada correlação positiva entre as características (objeto de discussão no próximo tópico).

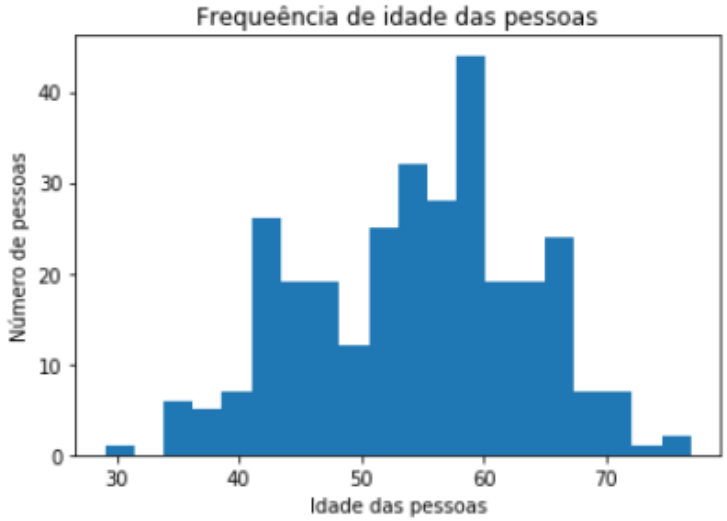
Quanto às estatísticas do conjunto de dados foi possível observar que:

- A maior parte dos participantes (68,32%) era do sexo masculino;
- A média de idade dos participantes foi de 54 anos de um conjunto que variou entre 29 e 77 anos de idade;
- A média de açucar no sangue em jejum apresentou um resultado de cerca de 15%;
- Em 33% dos participantes foi constatada a existência angina induzida por exercício;
- A média da frequência cardíaca máxima alcançada foi de 149.64 (variou entre o mínimo de 71 e o máximo de 202).

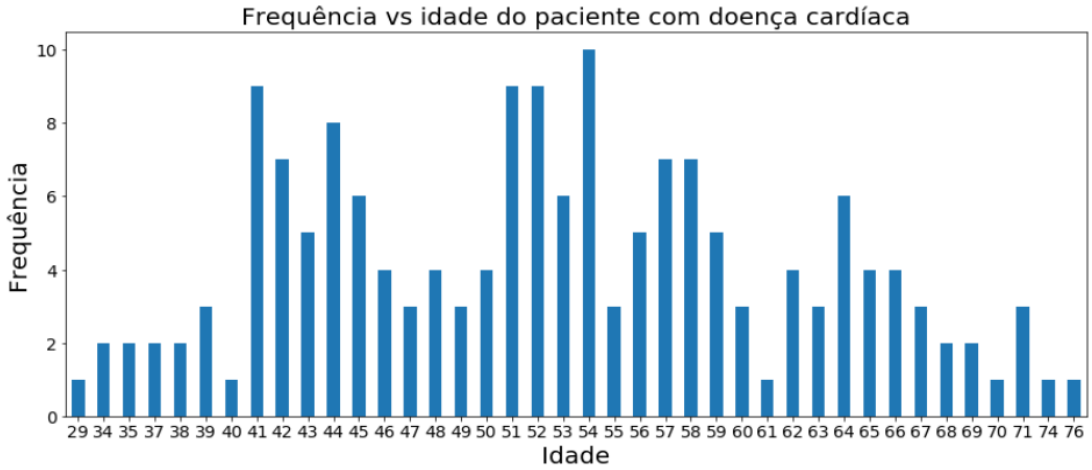
Visualização Exploratória

Durante a execução do projeto foram realizadas várias visualizações com objetivo tentar encontrar tendências que pudessem ajudar a prever eventuais eventos cardíacos, em especial, por meio da análise de correlação dos atributos.

Investigando a primeira característica “age” foi verificado que grande parte dos participantes estava na faixa etária entre 50 e 60 anos (média de 54 anos).

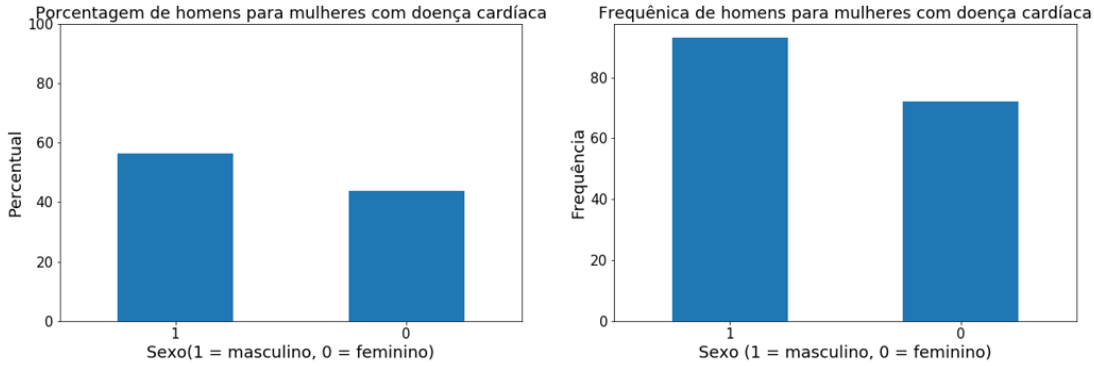


A frequência das pessoas com doença cardíaca ficou distribuída de forma semelhante.

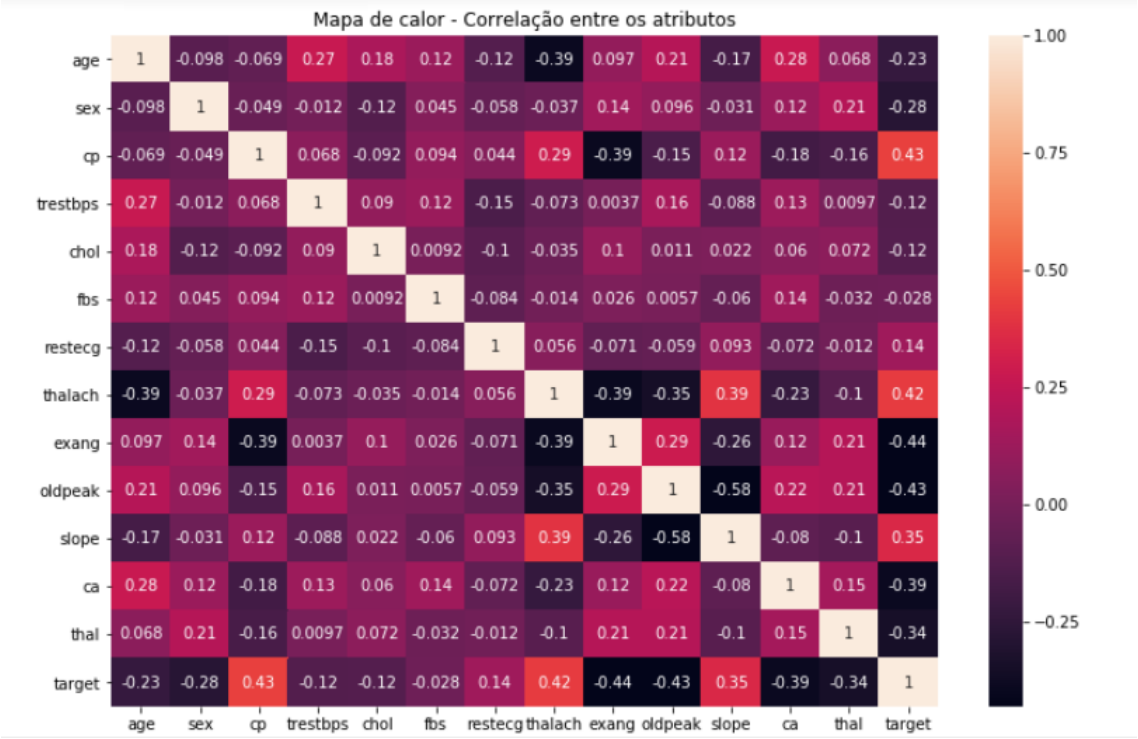


Analisando a característica sexo foi possível constatar que a maioria dos participantes da pesquisa eram do sexo masculino.

Essa tendência também foi mantida em relação às pessoas com doenças cardíacas, quando verificadas as frequências em relação ao sexo.

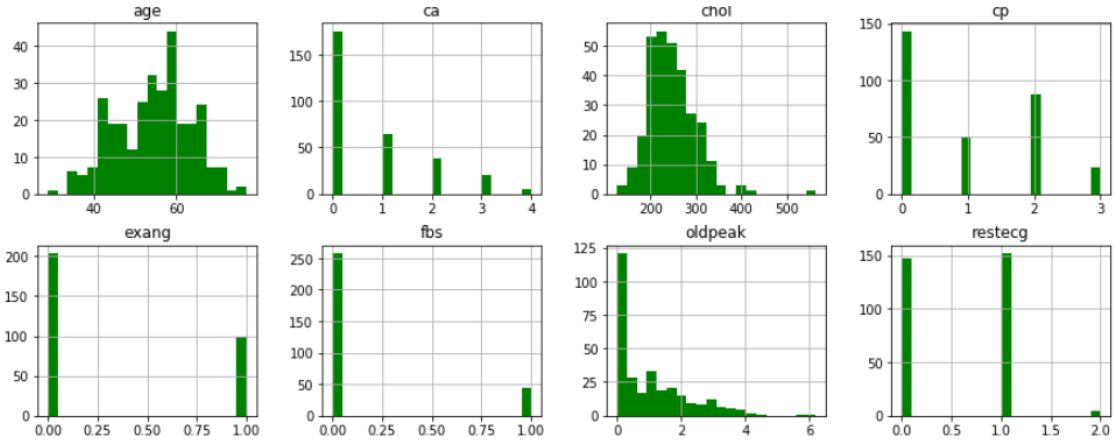


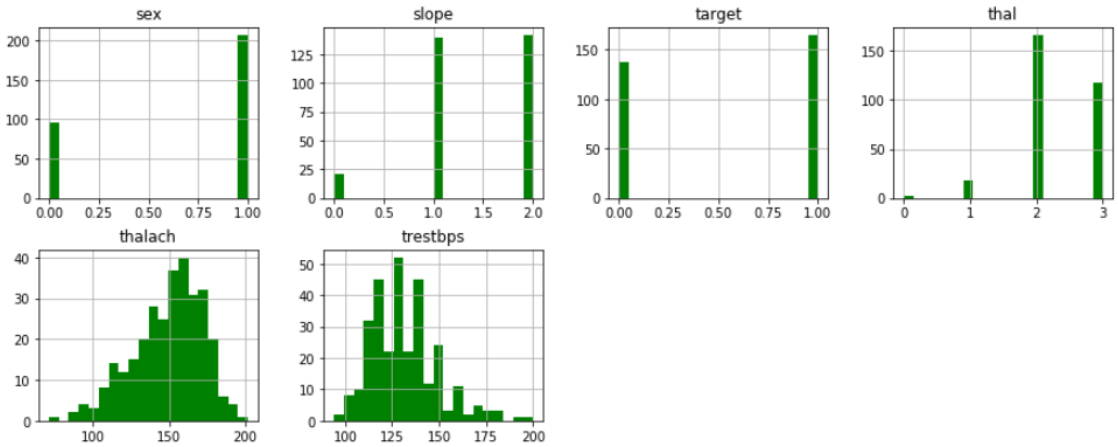
Tentando entender as correlações entre as características, principalmente em relação à variável alvo, foi gerado um mapa de calor.



Conforme o mapa de calor, mostraram alguma correlação positiva com a variável alvo as características: "cp" - chest pain type (tipo de dor no peito), "resteceg" - resting electrocardiographic results (resultados eletrocardiográficos em repouso(valores 0,1,2));, "thalach" - maximum heart rate achieved (frequência cardíaca máxima alcançada) e "slope" - the slope of the peak exercise ST segment (a inclinação do segmento ST de pico do exercício). A importância dessas correlações com as predições podem ser verificadas por ocasião da aplicação dos algoritmos escolhidos.

Ainda, com objetivo de uma visão geral acerca da forma da distribuição dos dados foi realizada uma plotagem de todas as características. Essa abordagem pode subsidiar decisões posteriores, como exemplo, a forma de tratamento dos outliers.





Algoritmos e Técnicas

Para esse projeto foram aplicados, num primeiro momento, três algoritmos de aprendizagem supervisionada de Machine Learning: **AdaBoost, Support Vector Machines (SVM)** e **Logistic Regression.** A escolha de modelos de aprendizagem supervisionada deve-se em razão da existência de um conjunto de dados com a variável alvo já definida para o treinamento.

A seguir seguem as características e a justificativas do uso dos modelos:

AdaBoost

Características do modelo

- O Classificador Adaboost é um algoritmo que faz parte da abordagem dos métodos de aprendizado Ensemble. Ele combina vários classificadores de baixo desempenho para que você obtenha um classificador forte de alta precisão. Uma das aplicações mais conhecidas do Adaboost é para sistemas de reconhecimento facial.
- O modelo AdaBoost é de fácil implementação e tem como vantagens: A correção iterativa dos erros do classificador fraco e a melhora da precisão por meio da combinação alunos fracos. Você pode usar muitos classificadores base com o AdaBoost. Esse modelo não é propenso a overfitting.
- O modelo AdaBoost é sensível a dados de ruído, sendo altamente afetado por outliers porque tenta encaixar cada ponto perfeitamente. O modelo pode ser considerado lento, dependendo da quantidade de dados, se comparado a outros.

Justificando o uso do modelo

Trata-se de uma base de dados pequena (eventual lentidão não será um problema) que contém dados categóricos e numéricos. O AdaBoost é um bom candidato, pois pode ser utilizado para executar várias iterações por ocasião do treinamento e ter menos erros no conjunto de dados de teste, tendo em vista sua capacidade de generalizar as tendências nos dados.

Support Vector Machines (SVM)

Características do modelo

- O SVM é um algoritmo binário de classificação muito utilizado na classificação de imagens. Devido suas vantagens, o SVM tem sido muito utilizado na construção de classificadores em vários campos, incluindo a microbiologia.

- Esse algoritmo consegue trabalhar bem com grandes conjuntos de exemplos e trata bem dados de alta dimensão.
- Para que ofereça um bom desempenho, as vezes, será necessária a definição de um kernel, o que pode ser meio trabalhoso. Dependendo do número de exemplos e da dimensionalidade dos dados o tempo de treinamento pode ser longo.

Justificando o uso do modelo

Verificou-se que o algoritmo é um bom candidato para resolução do problema por se tratar de um classificador binário muito eficiente. Além disso, a base de dados é pequena, então, eventual lentidão não será problema.

Logistic Regression

Características do modelo

- Algoritmo de classificação aplicado a problemas binários, bem utilizado no Mercado Financeiro (Análise de risco de crédito e fraudes em transações) e Administração Pública (Fraudes, evasão escolar e outros).
- O algoritmo de regressão é um modelo rápido no treinamento e na predição, sendo eficiente para conjuntos de dados pequenos e recursos limitados.
- O modelo é afetado negativamente por outliers. Dependendo do caso e dos recursos empregados o modelo pode sofrer sobreajuste.

Justificando o uso do modelo

A regressão logística é um modelo simples para separação entre duas classes que se encaixa na resolução do problema em questão.

Aplicação dos modelos

Considerando às características da base de dados (303 entradas, 14 atributos, dados numéricos), verifica-se que os modelos escolhidos podem obter bons resultados na predição. Num primeiro momento foram aplicados os modelos em sua forma padrão para análise dos resultados, visando uma posterior implementação do melhor modelo com otimização e aplicação de vários parâmetros.

Referência

Naive Predictor

Como modelo de referência foi utilizado um naive predictor que prediz o resultado sempre como 0 (não possui doença). O propósito ao gerar um naive predictor é simplesmente exibir como um modelo sem nenhuma inteligência se comporta em relação ao problema apresentado. Esse modelo serviu utilizado como base para comparação com os outros modelos posteriormente aplicados.

A métrica utilizada, nesse, caso para comparação foi a acurácia. Essa escolha se justifica, uma vez que as classes da variável alvo são balanceadas, sendo que essa situação não produz um resultado demasiadamente positivo e fora da realidade.

Fórmula utilizada: accuracy = (TP + TN) / (TP + FP + TN + FN)

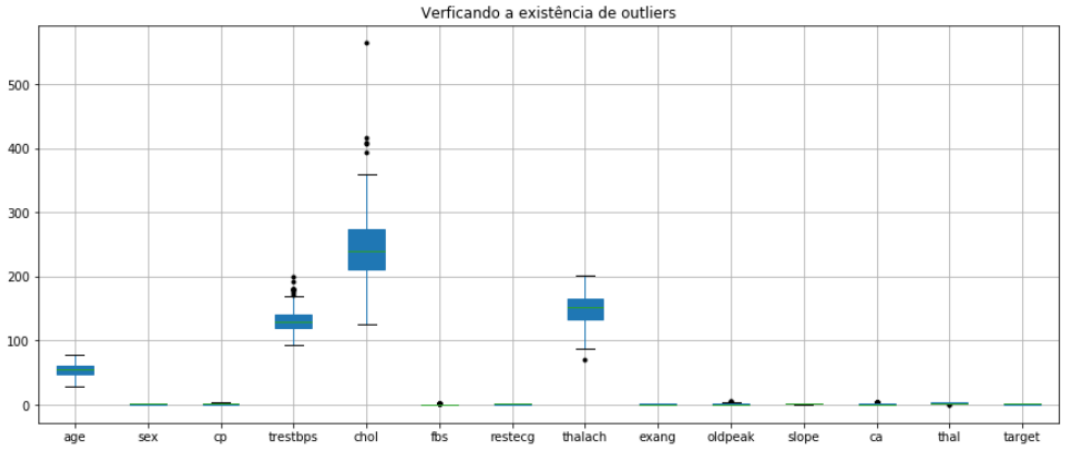
Resultado obtido: 0.4554

III Metodologia

Pré-processamento de dados

Tratamento dos outliers

Considerando que os modelos de aprendizado escolhidos são afetados negativamente pela presença de outliers foi realizada uma busca desses valores por meio de visualização.



Constatada a existência de valores discrepantes foi criada a função *outliers* para identificação específica de cada outlier.

Identificados os outliers e levando em consideração as características das distribuições dos dados nos atributos "trestbps", "chol" e "thalach", foi aplicada a mediana para o tratamento desses valores discrepantes.

Normalizando os dados

Considerando que a escala original dos dados pode afetar nos resultados de alguns algoritmos de Machine Learning como SVM (usado nesse projeto) e outros como KNN, nesse projeto foi realizada uma adaptação da escala de alguns dos valores numéricos (atributos: age', 'cp', 'trestbps', 'chol', 'restecg', 'thalach', 'oldpeak', 'slope', 'ca', 'thal'). Para tanto, foi utilizada a biblioteca ***sklearn.preprocessing.MinMaxScaler.***

Implementação

Dividindo os dados para treinamento e teste

Inicialmente, com os dados já prontos para utilização, foi realizada a divisão do conjunto de dados entre subconjuntos de treinamento e de teste, sendo que 80% dos dados foram separados para treinamento e 20% para teste. Para tanto foi utilizada a função ***train_test_split*** da biblioteca ***sklearn.model_selection.***

Aplicando Modelos de Aprendizado Supervisionado

Para avaliar adequadamente a performance de cada um dos modelos (AdaBoost, Support Vector Machines (SVM) e Logistic Regression) foi criado um pipeline de treinamento e predição para treinar os modelos, utilizando três tamanhos de conjuntos de dados para treinamento (10%, 50% e 100% das amostras), além de performar predições nos dados de teste.

Para tanto foi criada uma função para utilização durante a aplicação de treinamento e teste nos modelos de aprendizagem denominada *def train_predict(learner, sample_size, X_train, y_train, X_test, y_test).* Nessa função foram incluídas marcações para contagem de tempo *(função time da biblioteca time*) e scores *(funções accuracy_score, precision_score, recall_score, f1_score da biblioteca sklearn.metrics).* Os resultados foram direcionados para um dicionário denominado "results".

Após, por meio das bibliotecas *sklearn.ensemble, sklearn.svm e sklearn.linear*, respectivamente foram importados os classificadores AdaBoostClassifier, SVC e LogisticRegression.

Em seguida, por meio de um laço de repetição foram realizados os ajustes dos modelos de acordo com os dados de treinamento fornecidos e também as predições sendo que os resultados foram direcionados para "results".

Visualizando o dicionário results:

```
AdaBoostClassifier treinado em 24 amostras.
AdaBoostClassifier treinado em 242 amostras.
SVC treinado em 24 amostras.
SVC treinado em 242 amostras.
LogisticRegression treinado em 24 amostras.
LogisticRegression treinado em 242 amostras.
{'AdaBoostClassifier': {0: {'train_time': 0.07387852668762207, 'pred_time': 0.00997304916381836, 'acc_train': 0.789256198347107
5, 'acc_test': 0.7377049180327869, 'prec_train': 0.7898550724637681, 'prec_test': 0.7647058823529411, 'rec_train': 0.8320610687
022901, 'rec_test': 0.7647058823529411, 'f_train': 0.8104089219330856, 'f_test': 0.7647058823529412}, 1: {'train_time': 0.04391
2649154663086, 'pred_time': 0.009976625442504883, 'acc_train': 0.9297520661157025, 'acc_test': 0.8524590163934426, 'prec_trai
n': 0.9253731343283582, 'prec_test': 0.8787878787878788, 'rec_train': 0.9465648854961832, 'rec_test': 0.8529411764705882, 'f_tr
ain': 0.9358490566037737, 'f_test': 0.8656716417910447}}, 'SVC': {0: {'train_time': 0.014679193496704102, 'pred_time': 0.001994
1329956054688, 'acc_train': 0.7727272727272727, 'acc_test': 0.819672131147541, 'prec_train': 0.740506329113924, 'prec_test': 0.
7674418604651163, 'rec_train': 0.8931297709923665, 'rec_test': 0.9705882352941176, 'f_train': 0.8096885813148789, 'f_test': 0.8
571428571428571}, 1: {'train_time': 0.003964424133300781, 'pred_time': 0.003021240234375, 'acc_train': 0.8264462809917356, 'acc
_test': 0.819672131147541, 'prec_train': 0.7947019867549668, 'prec_test': 0.7948717948717948, 'rec_train': 0.916030534351145,
'rec_test': 0.9117647058823529, 'f_train': 0.851063829787234, 'f_test': 0.8493150684931507}}, 'LogisticRegression': {0: {'train
_time': 0.07903313636779785, 'pred_time': 0.00399610641479492, 'acc_train': 0.7892561983471075, 'acc_test': 0.786885245901639
3, 'prec_train': 0.803030303030303, 'prec_test': 0.8, 'rec_train': 0.8091603053435115, 'rec_test': 0.8235294117647058, 'f_trai
n': 0.806083650190114, 'f_test': 0.8115942028985507}, 1: {'train_time': 0.003991603851318359, 'pred_time': 0.00199246406555175
8, 'acc_train': 0.8347107438016529, 'acc_test': 0.8524590163934426, 'prec_train': 0.8226950354609929, 'prec_test': 0.8378378378
378378, 'rec_train': 0.8854961832061069, 'rec_test': 0.9117647058823529, 'f_train': 0.8529411764705882, 'f_test': 0.87323943661
97184}}}
```

Refinamento

Primeiramente, o dicionário results foi analisado. Após, foi escolhido o modelo que apresentou o melhor resultado etapa anterior:

Visulizando os resultados dos algortimos com 100% das amostras

Scores / Algoritmos	AdaBoostClassifier	Support Vector Machines -SVC	Logistic Regression
Acurácia no treinamento	0.9297520661157025	0.8264462809917356	0.8347107438016529
Acurácia no teste	0.8524590163934426	0.819672131147541	0.8524590163934426
Precisão no treinamento	0.9253731343283582	0.7947019867549668	0.8226950354609929
Precisão no teste	0.8787878787878788	0.7948717948717948	0.8378378378378378
Recall no treinamento	0.9465648854961832	0.916030534351145	0.8854961832061069
Recall no teste	0.8529411764705882	0.9117647058823529	0.9117647058823529
F1-score no treinamento	0.9358490566037737	0.851063829787234	0.8529411764705882
F1-score no teste	0.8656716417910447	0.8493150684931507	0.8732394366197184

O modelo que apresentou o melhor desempenho é o AdaBoostClassifier com os melhores resultados de acurácia e de precisão, também apresentando bons resultados no recall e no F1 - score. Os modelos Support Vector Machines (SVM) e Logistic Regression apresentaram bons resultados no recall e no F1-score. Devido aos resultados de melhor performance nos dados de teste (acurácia) e da característica de menor propensão a overfiting, bem como melhor precisão, acredito que o AdaBoostClassifier é o modelo mais apropriado para tarefa proposta nesse projeto.

Para o refinamento do modelo foi realizada uma busca em grade por meio da função *GridSearchCV* da biblioteca *sklearn.model_selection*.

Foram testados os seguintes parâmetros no classificador AdaBoostClassifier:

- '**base_estimator' :** O estimador de base foi o DecisionTreeClassifier;
- '**n_estimators':**O número máximo de estimadores em que o aumento é finalizado [30, 40, 50];
- '**learning_rate':** A taxa de aprendizado reduz a contribuição de cada classificador por learning_rate [.5, .8, 1];
- '**max_depth' (do base_estimator):** A profundidade máxima da árvore [2, 4, 6];
- '**min_samples_split' (do base_estimator):** O número mínimo de amostras necessárias para dividir um nó interno [6, 7];

O make_scorer escolhido foi o precision_score.

IV. Resultados

Avaliação e validação de modelos

Considerando que a base de dados era pequena (303 entradas), os modelos AdaBoost, Support Vector Machines (SVM) e Logistic Regression foram testados inicialmente com 10%, 50% e 100% das amostras, respectivamente, 24, 121 e 242 amostras. Todos eles apresentaram uma melhora nos resultados quando foi aumentado o número de amostras da base de dados.

Após o refinamento, o modelo AdaBoost apresentou os seguintes resultados com 100% das amostras:

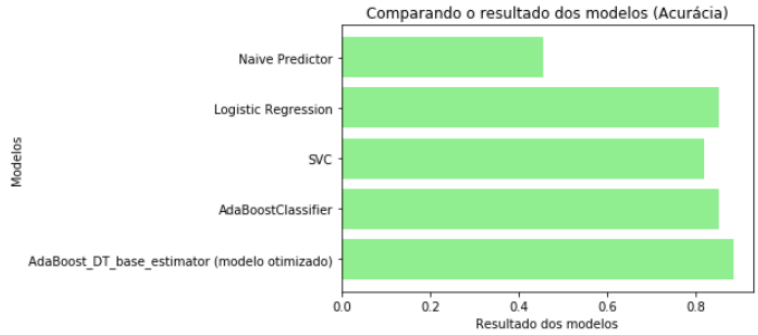
```
Modelo otimizado
------
Acurácia no teste: 0.8852
Precisão no teste: 0.9091
Recall no teste: 0.8824
F1-score no teste: 0.8955
```

Pode-se verificar que o modelo otimizado obteve os melhores resultados de acurácia, precisão e F1-score, quando comparado a todos os modelos anteriormente testados.

Considerando que o modelo escolhido tem uma menor propensão ao sobreajuste, acredito que alterações aumentando a base de dados não afetariam de forma negativa o modelo escolhido.

Justificação

Conforme o gráfico abaixo é possível comparar os resultados de todos os classificadores testados (métrica utilizada acurácia).



Em relação às predições é possível observar que o modelo otimizado (AdaBoostClassifier) obteve um resultado bem melhor do que o modelo de referência, Naive Predictor, e que os demais modelos testados, Support Vector Machines (SVM) e Logistic Regression.

Assim, considerando que em todas as métricas analisadas o modelo otimizado obteve um resultado de aproximadamente 90%, é possível inferir que o projeto atendeu às expectativas.

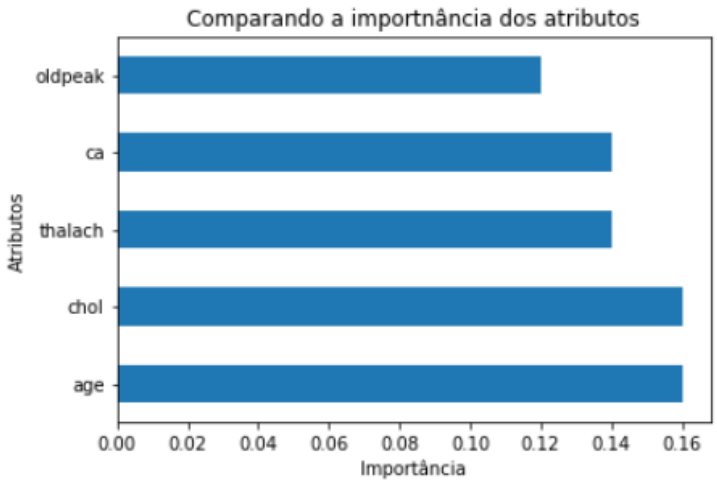
V. conclusão

Visualização de forma livre

Importância das features

Outra informação interessante para se constatar é a importância de cada atributo dos registros do conjunto de dados quando realizamos predições baseadas no algoritmo escolhido (AdaBoost_DT_base_estimator (modelo otimizado)).

Para tanto foi utilizado o recurso *feature_importance_* , tendo como resultado os 5 atributos mais importantes para o classificador escolhido.



Essa é uma informação interessante para verificar a importância dos atributos e compará-la com a relação entre essas características na busca de eventuais tendências no conjunto dos dados.

Reflexão

Em busca de soluções para o problema em questão o projeto seguiu a seguinte sequência de atividades:

- Exploração nos dados para verificar as características como: tipos de valores existentes (numéricos: (int, float,..), strings, time,...); a consistência dos dados (qualidade e arrumação); relação entre os atributos; existência de outliers, também de dados faltantes; verificar a necessidade de normalização de atributos numéricos; entre outros ajustes entendidos como necessários.

- Pré-processamento dos dados com o tratamento dos outliers e a normalização dos atributos com dados numéricos;
- Divisão dos dados entre conjuntos de treinamento e de teste, sendo que 80% dos dados serão utilizados para treinamento e 20% para teste.
- Terminada a fase de preparação e divisão dos dados, foram aplicados algoritmos de machine learning na seguinte forma: primeiramente o modelo de referência e seus resultados; após foram testados os algoritmos **AdaBoost, Support Vector Machines (SVM) e Logistic Regression.** Esses algoritmos foram escolhidos devido suas caraterísticas em relação ao problema proposto.
- Durante o treinamento, também foi realizada uma etapa de otimização do modelo que obteve o melhor desempenho na etapa anterior através da busca dos parâmetros ótimos.
- Por fim foram apresentados os resultados obtidos.

Em conclusão ao presente projeto cabem as seguintes considerações:

a) Em relação à Análise dos dados, foi possível constatar:

- A base de dados possuia o total de 303 registros, sendo que em relação à variável alvo (possui doença) havia 165 resultados positivos e 138 resultados negativos, mostrando que as classes eram balanceadas (54.46% e 45,54%);
- Em relação à idade havia registros de pessoas com 29 a 77 anos. As distribuições dos números de pessoas em relação às pessoas com doença, bem como o mapa de calor não apresentaram correlação direta entre a variável alvo e a idade. Contudo, para o algoritmo que obteve o melhor desempenho a variável idade foi considerada uma das mais importantes;
- Em relação à característica sexo, o conjunto possuia o percentual de indivíduos do sexo masculino de 68.32% e do sexo feminino de 31.68%. Das informações extraídas nas imagens (mapa do calor e histogramas) não foi possível afirmar que esse atributo tinha forte relação com a variável alvo.
- Conforme o mapa de calor, mostraram correlação positiva com a variável alvo as características "cp" - chest pain type (tipo de dor no peito), "resteceg" - resting electrocardiographic results (resultados eletrocardiográficos em repouso(valores 0,1,2));, "thalach" - maximum heart rate achieved (frequência cardíaca máxima alcançada) e "slope" - the slope of the peak exercise ST segment (a inclinação do segmento ST de pico do exercício). Já para o algortimo de Machine Learning que obteve os melhores resultados nas predições as características mais importantes foram: idade, colesterol sérico, frequência cardíaca máxima alcançada, número de grandes vasos(number of major vessels (0-3) colored by flourosopy) e depressão do segmento ST induzida pelo exercício em relação ao repouso. Diante disso, é possível constatar que o atributo "frequência cardíaca máxima alcançada" teve grande influência nos resultados.

Em relação à classificação e predição com algoritmos de Machine Learning:

Em relação às predições foram utilizados um modelo de referência (Naive Predictor) e mais três algoritmos AdaBoost, Support Vector Machines (SVM) e Logistic Regression. Após, foi realizada a verificação dos resultados utilizando as seguintes métricas: acurácia, precisão, recall e f1-score.
Durante a verificação dos resultados foi constatado que o algoritmo que obteve o melhor resultado foi o AdaBoostClassifier.

Constatado o melhor resultado, foi realizada uma otimização dos parâmetros do algoritmo AdaBoostClassifier com a inclusão do classificador DecisionTreeClassifier como *base estimator*, bem como a inclusão de números de estimadores, taxas de aprendizagem e mais dois parâmetros vinculados ao estimador base.

O resultado final obtido pelo modelo otimizado foi:

- Acurácia no teste: 0.8852

- Precisão no teste: 0.9091
- Recall no teste: 0.8824
- F1-score no teste: 0.8955

Diante do exposto acima, é possível inferir que, apesar das dificuldades, principalmente em relação à escolha dos algoritmos utilizados, o presente projeto atingiu os resultados esperados, ressalvando que sempre é possível melhorar.

Melhoria

Acredito que uma melhoria para o presente projeto poderia ser a utilização/comparação de outros algoritmos de Machine Learning (exemplos: Gaussian Naive Bayes (GaussianNB), Ensemble Methods (Random Forest, Gradient Boosting), K-Nearest Neighbors (KNeighbors), Stochastic Gradient Descent Classifier (SGDC)).