Projeto de Engenharia do Conhecimento

Realizado pelo Grupo 32:

- Gabriel Henriques 58182: 35 horas

- Guilherme Sousa 58170: 30 horas

Introdução e Objetivos

Este relatório apresenta os resultados de uma análise de um conjunto de dados relacionado à tireoide, especificamente uma versão do conjunto de dados "thyroid0387".

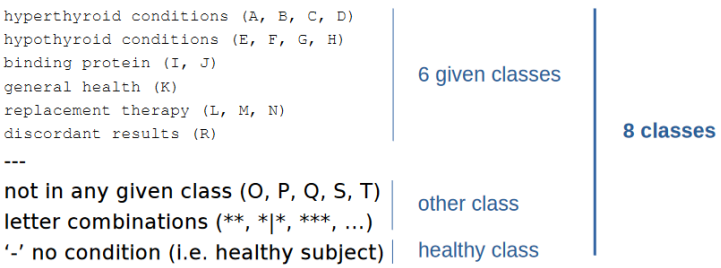
Foi-nos dado como objetivo principal deste trabalho desenvolver o melhor modelo de classificação possível de modo a prever o diagnóstico de pacientes com base nas características disponibilizadas no dataset, e categorizá-las em oito classes de diagnóstico distintas. Também a partir dos mesmos dados, tivemos de desenvolver modelos para prever a idade e o sexo dos pacientes.

Dos modelos obtidos, analisámos também quais as características que mais influenciaram a sua previsão.

Ao longo deste relatório, descreveremos os passos e métodos que usámos para processar os dados, selecionar as variáveis, selecionar os modelos e ajustar os parâmetros dos modelos.

Processamento de Dados

No passo de processamento, fizemos inicialmente uma limpeza dos dados onde separámos as colunas do dataset em dois tipos: características categóricas e características contínuas, substituímos todos os dados em falta (com '?') por NaNs, removemos a coluna "[record identification]" e traduzimos a coluna "diagnoses" da seguinte forma:



No caso da previsão do sexo, removemos ainda as linhas do dataset que tinham a característica “sex” em falta. Posteriormente, corremos uma pipeline com métodos de pré-processamento separados para as variáveis categóricas e contínuas.

Nas categóricas é executado um SimplerImputer com a estratégia “constant” que substitui todos os NaNs por “missing” e um OrdinalEncoder que, devido à limitação do scikit-learn, é necessário para converter as variáveis categóricas para numéricas. Escolhemos este encoder pois, apesar de não acharmos que exista necessariamente uma ordem nas variáveis categóricas, encontrámos dificuldades em usar outros dois encoders que acharíamos ser mais corretos (OneHotEncoder e LabelEncoder).

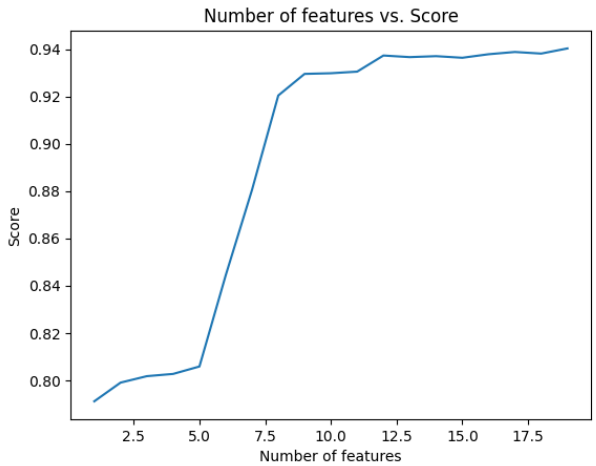
Nas variáveis contínuas é executado também um SimpleImputer com a estratégia “constant” que substitui os NaNs por “-1” para diferenciar os dados não medidos, uma vez que valores negativos não são naturais em dados de saúde, e um MinMaxScaler pois como existem variáveis de grandezas bastante diferentes foi necessário escalar de modo que essa diferença de grandeza não influencie o seu peso.

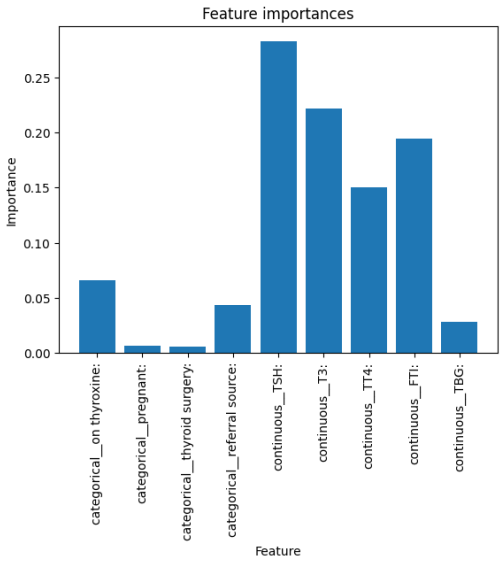
Fizemos a decisão de não substituir os dados em falta por outros valores, usando a média ou valor mais frequente por exemplo, pois sendo dados de saúde medidos, esses valores substitutos poderiam não refletir a verdadeira condição de saúde do paciente. Além disso, a ausência de dados pode em si ser informativa e indicar certas condições de saúde ou padrões de coleta de dados.

Seleção de variáveis

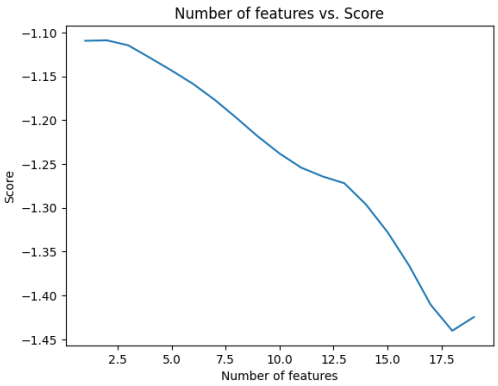
Para a seleção de variáveis, usámos o SequentialFeatureSelector com o modo default de seleção (forward selection) e como modelos, para a previsão de diagnóstico e sexo, usámos o DecisionTreeClassifier, e para a previsão da idade usámos o LinearRegression, por serem os modelos em que estamos mais familiarizados. Nos casos em que o score se manteve similar, optámos por selecionar o menor número de atributos de modo a evitar o risco de overfiting aos dados de training.

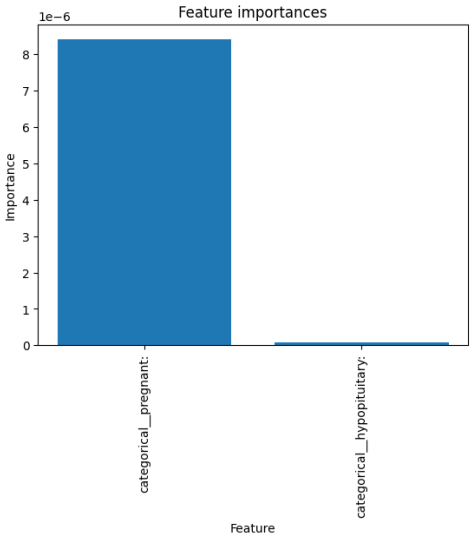
Previsão de Diagnóstico:



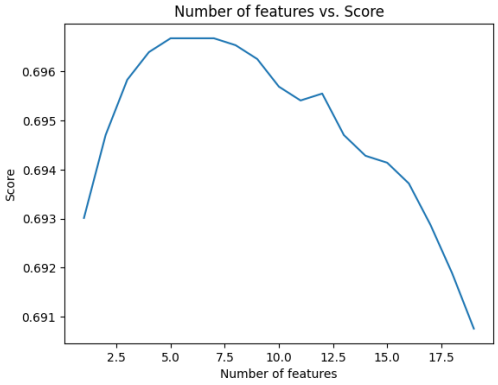


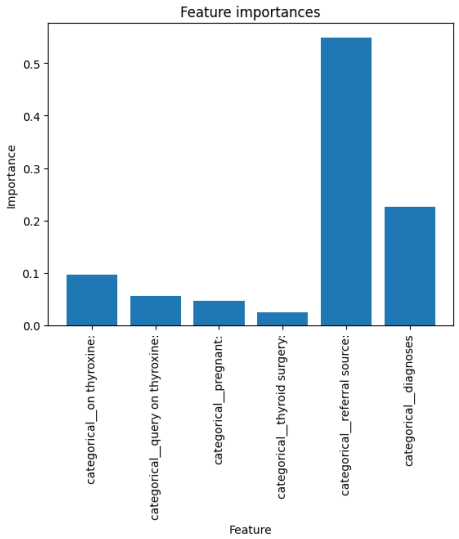
Previsão de Idade:





Previsão de Sexo





Resultado dos Modelos

Testámos diversos modelos, dos quais já eramos familiares devido às aulas práticas, com cross validation de modo a ver qual se adequava melhor aos dados e obtivemos os seguintes resultados:

Previsão de Diagnóstico:

```
Cross validation score for model DecisionTreeClassifier() is 0.9306319408457453
Cross validation score for model GaussianNB() is 0.14118397664142474
Cross validation score for model KNeighborsClassifier() is 0.8612661105631801
Cross validation score for model SVC() is 0.7451618625194794
Cross validation score for model RandomForestClassifier() is 0.941261894280901
```

Previsão de Idade:

```
Cross validation score for model DecisionTreeRegressor() is -1.1089895837682484
Cross validation score for model RandomForestRegressor() is -1.2372806419018503
Cross validation score for model KNeighborsRegressor() is -0.07353986645467354
Cross validation score for model SVR() is -0.0046267759322911935
Cross validation score for model LinearRegression() is -1.1089895837682473
```

Previsão de Sexo:

```
Cross validation score for model DecisionTreeClassifier() is 0.6966774856326117
Cross validation score for model GaussianNB() is 0.3388265888495171
Cross validation score for model KNeighborsClassifier() is 0.6559573792295705
Cross validation score for model SVC() is 0.6830092606378226
Cross validation score for model RandomForestClassifier() is 0.6942821268697456
```

Ajuste de Hiper parâmetros

Dos modelos testados anteriormente, selecionámos os que obtiveram melhor resultado para ajustar os seus parâmetros de modo a obter resultados ainda melhores! Para tal usámos o GridSearchCV com o default 5-fold cross validation devido à sua robustez, escalabilidade para com grandes conjuntos de dados e fácil utilização. Para cada previsão ajustámos os parâmetros e obtivemos os seguintes resultados:

Previsão de Diagnóstico:

```
decision_tree_params = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [3, 5, 10, 15],
    'min_samples_split': [2, 5, 9],
    'min_samples_leaf': [1, 2, 5, 9],
    'ccp_alpha': [0.0, 0.0001, 0.001, 0.01]
}
dt = fine_tune_model(X_train_selected, y_train, DecisionTreeClassifier(), decision_tree_params)

random_forest_params = {
    'n_estimators': [50,100],
    'criterion': ['gini', 'entropy'],
    'max_depth': [3, 5, 10, 15],
    'min_samples_split': [2, 5, 9],
    'min_samples_leaf': [1, 2, 5, 9],
    'ccp_alpha': [0.0, 0.0001, 0.001, 0.01]
}
rf = fine_tune_model(X_train_selected, y_train, RandomForestClassifier(), random_forest_params)
```

```
Best cross-validation score: 0.9363548475173156
Best estimator: DecisionTreeClassifier(ccp_alpha=0.001, criterion='entropy', max_depth=15,
                       min_samples_leaf=2)

Best cross-validation score: 0.9430343116222657
Best estimator: RandomForestClassifier(ccp_alpha=0.0001, criterion='entropy', max_depth=15,
                       min_samples_split=5)
```

Previsão de Idade:

```
svr_params = {
    'C': [0.1, 1, 10, 100],
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'degree': [2, 3, 4, 5],
    'gamma': ['scale', 'auto']
}
svr = fine_tune_model(X_train_selected, y_train, SVR(), svr_params)
```

```
Best cross-validation score: -0.004428228451638127
Best estimator: SVR(C=1, degree=2, gamma='auto')
```

Previsão de Sexo:

```
decision_tree_params = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [3, 5, 10, 15],
    'min_samples_split': [2, 5, 9],
    'min_samples_leaf': [1, 2, 5, 9],
    'ccp_alpha': [0.0, 0.0001, 0.001, 0.01]
}

dt = fine_tune_model(X_train_selected, y_train, DecisionTreeClassifier(), decision_tree_params)

random_forest_params = {
    'n_estimators': [50,100],
    'criterion': ['gini', 'entropy'],
    'max_depth': [3, 5, 10, 15],
    'min_samples_split': [2, 5, 9],
    'min_samples_leaf': [1, 2, 5, 9],
    'ccp_alpha': [0.0, 0.0001, 0.001, 0.01]
}

rf = fine_tune_model(X_train_selected, y_train, RandomForestClassifier(), random_forest_params)
```

```
Best cross-validation score: 0.6966774856326117
Best estimator: DecisionTreeClassifier(max_depth=10)

Best cross-validation score: 0.6975228538248518
Best estimator: RandomForestClassifier(max_depth=10, min_samples_leaf=2, min_samples_split=5,
                       n_estimators=50)
```

Discussão e Conclusões

Desta análise tirámos que para a previsão de diagnóstico o melhor modelo será o RandomForestClassifier com os parâmetros [ccp_alpha=0.0001, criterion='entropy', max_depth=15,min_samples_split=5], este obteve muito bons resultados em cross validation (0.943), e com o excerto dos dados disponibilizados para teste conseguiu uma accuracy de 0.85, ficámos, portanto, satisfeitos e com confiança que avaliará maioritariamente bem mais dados de teste que lhe sejam submetidos.

Já sobre a questão de prever a idade e sexo, assumimos poder usar como atributo os diagnósticos dos pacientes, que se vieram a provar relevantes na determinação do sexo, tendo sido a segunda característica com maior peso.

Na previsão do sexo, os resultados já não foram tão famosos quanto os do diagnóstico, tendo obtido em cross validation um score de 0.697, acabámos por escolher o modelo RandomForestClassifier com os parâmetros [max_depth=10, min_samples_leaf=2, min_samples_split=5,n_estimators=50], no entanto o modelo DecisionTreeClassifier obteve resultados muito similares. Concluímos então que podemos prever o sexo de um paciente com sensivelmente 70% de certeza.

Por fim, os modelos para prever a idade obtiveram resultados desastrosos, não conseguimos entender se por erro de nossa análise e interpretação ou se pela qualidade dos dados, no entanto mesmo assim conseguimos escolher e ajustar um modelo que tivesse resultados menos maus: SVR com os parâmetros [C=1, degree=2, gamma='auto']. Concluímos então que não é possível prever a idade de um paciente dados os outros atributos.