Grupo 10

- André Santos Nº53323: 24h
- Filipe Santos Nº53304: 24h
- João Martins Nº62532: 26h
- Rúben Torres Nº62531: 24h

Aprendizagem Automática 2023/2024

Third Home Assignment

Objective: Produce the best regression model for the following dataset;

Sumário do trabalho realizado:

1. Carregámos o dataset a partir do ficheiro: **"drd2_data.pickle"**. O **"*Data Splitting*"** está incorporado na operação de carregamento dos dados;
2. Fizemos scale dos dataframes que têm as variáveis independentes: **"X_train"** e **"X_ivs"**;
3. Fizemos ***"Feature Selection"***: através de análise de correlação e através de ***"Random Forest"*** com estimativa de importância;
4. Usámos o ***"Principal Component Analysis (PCA)"*** também no tratamento dos dados numa tentativa de retirar features redundantes ou que não fossem significativas para as previsões no dataset;
5. Aplicámos as implementações dos modelos de regressão do ***"scikit-learn"***: ***"Random Forest Regression"*** e ***"Linear Regression", "Decision Tree Regressor", "Support Vector Regression"***, entre outros;
6. Utilizámos ***"Cross-validation"*** de forma avaliar os modelos;
7. Escolhemos os melhores métodos de tratamentos de dados, assim como os melhores modelos de regressão através das métricas de avaliação apropriadas obtidas por ***"Cross-validation"***;
8. Efetuámos as previsões a entregar sobre **"X_ivs"** utilizando o melhor modelo identificado através de ***"Cross-validation"***;

Data Processing

Para todo o pipeline de ***"Data Processing"*** seguimos sempre a uma ordem fixa:

1. ***"Data Scaling"***;
2. ***"Feature Selection"***;
3. ***"PCA"***;

Ou seja, sempre que um destes passos é realizado, deve respeitar a ordem apresentada.
Nenhuma operação de imputação foi realizada, pois não havia qualquer dado em falta.
Sabendo que mais de 90% dos dados são 0 (valor numérico), escolhemos não considerar nenhum dos valores restantes como outliers (mesmo os valores mais elevados, mais de 500).
Vários métodos de scaling foram testados em combinação com a aplicação do ***"PCA"*** para diferentes números alvo de componentes para os diferentes modelos de regressão:

- ***"Standard Scaler"***;
- ***"Power Transformer"***;
- ***"Quantile Transformer (Gaussian e Uniform)"***;
- ***"Normalizer"***;

Os valores das métricas resultantes mostram uma clara superioridade do ***"Uniform Quantile Transformer"*** universal a todos os modelos, no entanto como não foi usado em aulas optámos por não usar para avaliação do modelo final. Acabando por ser escolhido o ***"Standard Scaler"***.

- André Santos Nº53323: 24h
- Filipe Santos Nº53304: 24h
- João Martins Nº62532: 26h
- Rúben Torres Nº62531: 24h

O dataset no seu estado inicial tem mais de 7 mil linhas e mais de 2000 mil colunas (*features*). Através de vários métodos de ***"Feature Selection"*** e ***"Dimensionality Reduction"*** procuramos diminuir substancialmente o número de *features*.

Feature Selection:

Os seguintes métodos de feature selection foram aplicados ao dataset de forma a selecionar um nº de colunas mais reduzido:

- Através de análise de correlação entre cada uma das variáveis independentes (features, em "X_train") e a variável independente (em "y_train"), permanecendo apenas as features com correlação superior ao threshold com a variável dependente (target). Vários valores de threshold foram testados sendo que visam cortar um nº de features moderado, nem demais nem de menos. Por fim escolhemos um threshold de 0.05 que resulta em 374 features.
- Stepwise forward feature selection que aplicamos, "SequentialFeatureSelector" do sklearn apresentou um tempo de execução demasiado longo para usar neste problema.
- Através da utilização de "RandomForest Regressor" para estimar a importância das features.
  Testámos vários valores de threshold de forma a fazer uma seleção de apenas as features com importância acima de um certo valor. Utilizando um "threshold" de 0.001 resultou um novo dataset com apenas 143 features.
- A combinação de correlation score feature selection seguida de random forest feature selection assim como a combinação destes com o pca (50 componentes) deu resultados bastante razoáveis, tendo em conta que o número de features do dataset resultante destas operações diminuiu drasticamente para só 50, mas não alterou drasticamente a performance dos modelos testados.
  X_train_cut_rffs_pca(already scaled!) running on random forest model
  The RVE is: 0.4000572834623738

Os features mais relevantes de acordo com o resultado do random forest feature selection são:

[0 1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39 40 41 42 …]

A grande maioria destas features também consta na lista de features escolhidas pelo correlation score.

Principal Component Selection (PCA)

Através de PCA é possível transformar e reduzir o nº de features do dataset. Deve-se ter cautela na escolha do nº componentes resultante, pois é importante ter em conta o nº de features já existentes. Foi este o método de component selection que mais explorámos. O PCA foi testado com um conjunto de diferentes valores de "n_components" sobre diferentes conjuntos de dados, cada um resultante da aplicação de cada um dos diferentes scalers aos

dados iniciais, os conjuntos de dados produzidos neste processo foram sujeitos um a um a utilização e posterior análise modelo a modelo de regressão. Os resultados ilustram que para a maioria dos modelos mais bem sucedidos o intervalo de “n_components” de [100, 200] é normalmente o ideal. O modelo SVR (Suport Vector Regressor) obteve o melhor valor de RVE antes de estabilização no valor de 200 “n_components”.

A aplicação de KPCA de 50 componentes com kernel polinomial gerou resultados ligeiramente inferiores aos resultados produzidos a partir de modelos sujeitos a pca convencional, reduzindo substancialmente o desempenho dos modelos;

A aplicação de KPCA de 50 componentes e kernel rbf gerou resultados em linha com os resultados do dataset com o grupo de controlo (X_train original sem qualquer tratamento) no caso de dados escalados, salientando o melhor resultado com pequenas melhorias de RVE com 0.4652, RMSE de 0.203, correlation Score de 0.6827, máximo erro de 0.82 e Mean absolute error de 0.1589 para o modelo Random Forest. No caso de dados não escalados, usando o KPCA, os resultados de cross-validation são verdadeiramente abismais exibindo valores das pontuações muito fracas.

Cross validation

O método principal usado na avaliação dos modelos e diferentes datasets que os treinaram. - Diferentes combinações de utilização ou ausência de scaling com métodos de “feature selection” e “principal component analysis” foram testados através de k-fold cross-validation e sujeitos a análise das respectivas métricas de avaliação. Para propósitos de análise foi mantido como “grupo de controlo”, correspondente ao dataset no seu estado inicial sem qualquer tratamento.

O Dataset escolhido

As métricas de avaliação do dataset foram inicialmente todas produzidas pela utilização do mesmo modelo (RandomForestRegressor), proporcionando base sólida de comparação de resultados.
Após a análise generalizada a mais modelos acabámos por usar o dataset resultante do standard scaler, com PCA pois, primeiro é mais acessível computacionalmente tendo demorado menos tempo a processar a operação sobre o dataset e as previsões resultantes deste dataset em comparação com outros datasets como o feature selection, usando o mesmo modelo, foram melhores com menor número de features.

Model tuning

Moder tuning foi experimentado em alguns modelos. Mas a aplicação dos melhores parâmetros sugeridos raramente resultou sequer em alguma melhoria. Optámos por

favorecer os parâmetros por defeito dos modelos, evitando assim por completo o overfitting por causas de overtuning dos modelos.

Avaliação dos modelos

Os melhores modelos e os respectivos valores das métricas de avaliação são apresentados na tabela abaixo.
Estes modelos foram treinados usando o nosso dataset escolhido: Standard Scaler com PCA 100 componentes; avaliados usando K-Fold Cross Validation.

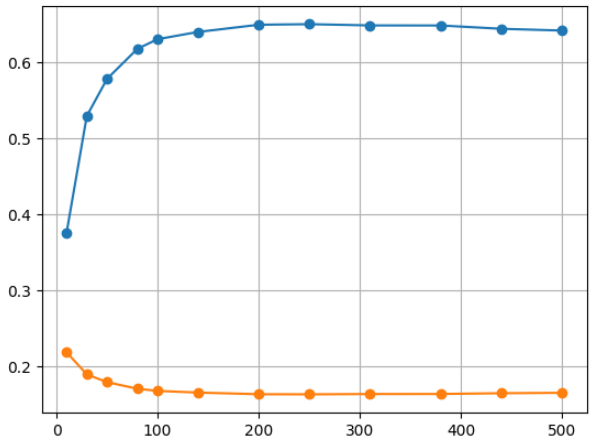

fig1: Gráficos que mostram o resultado do modelo SVR. Eixo x é o número de componentes e o eixo RVE linha azul e o RMSE é linha laranja. A linha azul representa evolução do RVE com o aumento do número de componentes do dataset. A linha laranja representa evolução do RMSE com o aumento do número de componentes do dataset.
Tabela com modelos testados e respetivas estatísticas relevantes(Resultados):

Models	RVE	RMSE	Correlation Score	Maximum Error	Mean Absolute Error
SVR	0.6265	0.1691	0.7917	0.8692	0.1281
KNN	0.6032	0.1743	0.7792	0.9546	0.1257
LinearR	0.4026	0.2138	0.6347	0.847	0.1689
XGB	0.5897	0.1783	0.7723	0.8378	0.1317
Neural networks	0.3733	0.219	0.675	1.0536	0.1641

Grupo 10

- André Santos Nº53323: 24h
- Filipe Santos Nº53304: 24h
- João Martins Nº62532: 26h
- Rúben Torres Nº62531: 24h

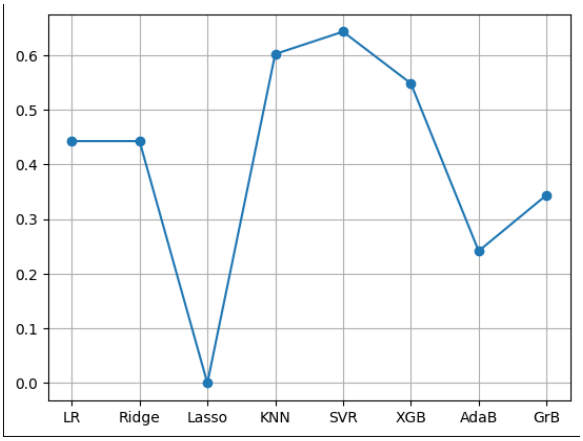

fig2:Gráfico que demonstra o RVE dos vários modelos treinados com o dataset standard scaled com pca 150 componentes.

Discussão e conclusões:

O processamento de dados não produziu melhorias significativas na qualidade das previsões dos modelos. No entanto é importante notar que a remoção de informação desnecessária no *dataset* resultou, de qualquer das formas, em melhorias nas previsões relativamente ao dataset inicial segundo a nossa cross-validation.
A redução muito acentuada de features quer pela definição de thresholds inflexíveis em "*Feature Selection*" quer pela escolha de nº de componentes muito baixa em PCA, ou combinação das duas muitas vezes não só se traduz em diferenças minutas como em redução da qualidade das previsões.
Selecionamos o modelo SVR treinado com o dataset produzido através do processo de standard scaling seguido de PCA com 100 componentes por apresentar os melhores resultados em todas as métricas, parecendo desse modo o modelo mais promissor para estes dados. A escolha de 100 componentes no pca deveu-se ao facto de garantir melhores resultados com um bom ratio nºcomponentes/RVE. Valores acima de 100 componentes já não representam uma grande melhoria no modelo final, como é possível verificar na figura1.

Link para repositório assignment3 grupo10:
https://github.com/rubentorres-developer/ML-ThirdHomeAssignment