Métodos de Regressão
Aprendizagem de Máquina
2024

Caio Shimohiro, Gustavo Hanke

1. Introdução

Este documento tem como objetivo descrever os processos de implementação e realizar um estudo e comparação entre os métodos de regressão apresentados na matéria de Aprendizagem de Máquina - Regressão Linear Múltipla (sem aprendizagem de máquina), os métodos monolíticos: KNN (KNR em regressão), SVM (SVR em regressão), MLP (Multilayer Perceptron) e os métodos de múltiplos regressores: Random Forest e Gradient Boosting. Todos aplicados em uma base de dados fornecida pelo professor e que será apresentada na sequência.

Vale ressaltar que todo o processo foi realizado na linguagem Python fazendo utilização sobretudo da biblioteca Scikit-Learn.

2. Implementação

2.1. Análise descritiva dos dados

A base de dados fornecida pelo professor e disponibilizada no site Kaggle apresenta um conjunto de 1338 instâncias compostas por 6 atributos (7 contando a saída) que variam entre valores numéricos e categóricos. Todos tem como objetivo descrever o perfil fisiológico de uma pessoa para realizar uma previsão de custo a ser cobrado por um plano de saúde. Os detalhes de cada um dos atributos segue:

Numéricos:

Atributo	**Tipo**	**Descrição**	**Valor mínimo**	**Valor máximo**	**Valor médio**
Idade	Inteiro	Idade do beneficiário	18	64	39,2
IMC	Real	Índice de Massa Corpórea do indivíduo (proporção altura x massa)	15,96	53,13	30,66
n° de Filhos	Inteiro	Número de filhos do beneficiário	0	5	1,09
Valor de cobrança	Real	Custo do plano para o beneficiário	1.121,87	63.770,42	13.270,42

Categóricos:

Atributo	Descrição	Distribuição
Sexo	Sexo do beneficiário: masculino ou feminino	• Masculino: 676 • Feminino: 662
Fumante	Paciente fumante ou não	• Sim: 274 • Não: 1064
Região	Área residencial do paciente nos EUA	• Sudoeste: 325 • Sudeste: 364 • Noroeste: 325 • Nordeste: 324

Observa-se que é uma base relativamente balanceada na sua distribuição de dados categóricos.

É preciso destacar que foram realizadas modificações para “eliminar” os valores categóricos para maior eficiência na execução do trabalho. No caso todos os valores foram substituídos por equivalentes numéricos de mesma representatividade.

O próximo passo na análise é verificar a correlação entre cada um dos atributos utilizando o índice de correlação de Pearson. Foi realizado o cálculo entre todos os atributos combinados 2 a 2 e seguem os resultados:

	Sexo	IMC	n° de Filhos	Fumante	Região	Cobrança
Idade	-0,020855872118	0,1092718815	0,042468998556	-0,025018751514	-0,003243435443	0,2990081933
Sexo		0,046371150665	0,017162977775	0,076184816912	-0,007974138241	0,0572920622
IMC			0,012758900882	0,003750425905	-0,1566856316	0,1983409688
n° de Filhos				0,007673120308	0,001907154144	0,06799822685
Fumante					-0,013246298882	0,7872514305
Região						-0,011740854881

Analisando os resultados levando em conta sobretudo a correlação dos demais atributos com o atributo de valor de cobrança, nota-se que o atributo mais correlato é o de fumante. Após ele, idade e IMC também são relevantes, porém não no mesmo nível.

Já atributos de região, número de filhos e sexo tiveram mínima correlação.

Ainda sobre o atributo de região, é difícil de levar em consideração o cálculo do coeficiente de correlação já que o dado não é adequado para uma análise matemática como essa já que seria necessário considerar uma região “maior” que a outra de alguma maneira para dar sentido aos números que as representam.

2.2. Divisão do conjunto de dados

A divisão dos dados foi realizada como solicitado na definição do trabalho, sendo 50% da base original destinada para o treinamento da base, 25% para validação e os outros 25% para testes.

Para garantir a aleatoriedade em cada iteração, foi utilizado a função shuffle() também do Sklearn no início de cada novo ciclo.

2.3. Treinamento e calibração dos modelos

Na fase de treinamento dos modelos, foram explorados diversos valores dos hiperparâmetros requisitados para cada método utilizando Grid Search com Cross Validation. Serão descritos a seguir cada um deles.

Lembrando que a métrica utilizada para a seleção dos parâmetros foi o de Raiz de Erro Médio Quadrático.

2.3.1. KNN

Para o KNN, o K, ou número de vizinhos, foi testado em uma faixa de 2 até 20. Já o weight (peso) foi variado entre os modos uniform, sem ponderar as distâncias, e distance, ponderando as distâncias.

Abaixo encontram-se os hiperparâmetros selecionados após treinamento e validação em cada iteração:

Iteração	K	Weight
1	20	distance
2	7	distance
3	6	distance
4	18	distance
5	19	distance
6	19	distance
7	16	distance
8	17	distance
9	8	distance
10	15	distance
11	11	distance
12	7	distance
13	5	distance
14	16	distance
15	19	distance
16	17	distance
17	13	distance

18	15	distance
19	19	distance
20	19	distance

Em análise, a média dos valores de K foi de 16. Enquanto o único peso escolhido foi o de distance.

2.3.2. SVM

Para o SVM foram explorados os parâmetros de kernel, selecionados entre linear, polinomial, RBF e sigmoid. Além do valor C de nível de generalização, que variou de 1 até 20. Os resultados se encontram a seguir:

Iteração	**Kernel**	**C**
1	linear	20
2	linear	20
3	linear	20
4	linear	20
5	linear	20
6	linear	20
7	linear	20
8	linear	20
9	linear	20
10	linear	20
11	linear	20
12	linear	20
13	linear	20
14	linear	20
15	linear	20
16	linear	20
17	linear	20
18	linear	20
19	linear	20
20	linear	20

A análise aqui é bastante simples. Apenas a configuração de Kernel linear e valor C de 20 foi escolhida.

2.3.3. MLP

O número de iterações do método variou entre 150, 300, 500 e 1000. Para as camadas ocultas no MLP, foram testadas diferentes configurações que variam de 1 até 3 camadas com quantidades de neurônios de 10, 15 ou 20. Em relação à função de ativação, foram entre identity (sem transformação), logistic (sigmoide), tanh (tangente hiperbólica) e relu (rectified linear unit). Por fim, a taxa de aprendizado variou entre constant (constante), invscaling (inversamente proporcional às iterações) e adaptative (mantém constante enquanto desempenho não cai). Os resultados abaixo:

Iteração	nº de iterações	Ativação	Taxa de aprendizagem	Camadas ocultas
1	1000	relu	constant	(20, 20, 20)
2	1000	relu	adaptive	(20, 20, 20)
3	1000	relu	constant	(20, 20, 20)
4	1000	relu	constant	(20, 20, 20)
5	1000	relu	invscaling	(20, 20, 20)
6	1000	relu	invscaling	(20, 20, 20)
7	1000	relu	invscaling	(20, 20, 20)
8	1000	relu	adaptive	(20, 20, 20)
9	1000	relu	invscaling	(20, 20, 20)
10	1000	relu	adaptive	(20, 20, 20)
11	1000	identity	invscaling	(20, 20, 20)
12	1000	relu	adaptive	(20, 20, 20)
13	1000	relu	adaptive	(20, 20, 20)
14	1000	relu	invscaling	(20, 20, 20)
15	1000	relu	constant	(20, 20, 20)
16	1000	relu	invscaling	(20, 20, 20)
17	1000	identity	constant	(20, 20, 20)
18	1000	relu	adaptive	(20, 20, 20)
19	1000	relu	adaptive	(20, 20, 20)
20	1000	relu	invscaling	(20, 20, 20)

O número de iterações foi constante em 1000 iterações. Houveram alguns avisos de não convergência durante os testes, mas optou-se por não aumentar o número de iterações com receio de overfitting.

A configuração das camadas ocultas também foi constante, com três camadas de 20 neurônios cada. Para a função de ativação, apenas relu e identity foram optados por, sendo relu a mais constante por uma boa margem. Já para a taxa de aprendizagem, invscaling foi a mais frequente, com 8 vezes escolhida.

2.3.4. Random Forest

O número de estimadores para o método foi de 50, 100 ou 150. O Criterion foi entre squared_error (erro quadrático médio), absolute_error (erro absoluto médio), Poisson (dados de contagem positiva) e friedman_mse (erro quadrático médio de Friedman). Profundidade foi de 5 até 20; mínimo de amostras por divisão foi de 1 até 5 e mínimo por folha foi de 2 até 20.

Iteração	n° de Estimadores	Criterion	Profundidade máxima	Min. amostras p/ divisão	Min. amostras p/ folha
1	50	friedman_mse	5	7	3
2	100	poisson	5	12	5
3	150	absolute_error	5	2	3
4	50	absolute_error	10	17	3
5	100	absolute_error	10	12	3
6	100	absolute_error	5	2	5
7	100	friedman_mse	5	12	5
8	100	squared_error	5	2	1
9	150	friedman_mse	5	7	5
10	100	friedman_mse	5	2	3
11	50	squared_error	5	2	1
12	50	squared_error	20	17	5
13	50	absolute_error	10	17	5
14	100	absolute_error	20	7	3
15	50	absolute_error	20	12	5
16	50	squared_error	5	17	5
17	100	friedman_mse	5	2	5
18	50	squared_error	5	12	5
19	50	absolute_error	15	2	5
20	100	absolute_error	20	7	3

Todos os parâmetros variaram bastante. Agora para cada caso, nota-se:

- O número de estimadores ficou em uma média de 82.5
- Erro absoluto médio foi o mais comum com 9 iterações de Criterion
- A profundidade máxima média foi de 9,25
- A média do mínimo de amostras para divisão foi de 8.5
- E amostras por folha foi de 3.9

2.3.5. Gradient Boosting

O número de estimadores para o método foi de 20, 60 ou 100. A função de perda foi entre squared_error (erro quadrático médio), absolute_error (erro absoluto médio), Huber (combina os dois erros, quadrático e absoluto) e quantile (estimação de quantis). Profundidade foi de 5 até 20 de 5 em 5; mínimo de amostras por divisão foi de 1 até 5 e mínimo por folha foi de 2 até 20.

Iteração	n° de Estimadores	Taxa de aprendizagem	Profundidade máxima	Min. amostras p/ divisão	Min. amostras p/ folha	Função de perda
1	60	0.05	5	17	1	huber
2	60	0.2	5	17	3	huber
3	100	0.2	5	2	5	absolute_error
4	100	0.2	5	17	5	absolute_error
5	60	0.15	5	17	3	huber
6	60	0.15	5	17	3	absolute_error
7	20	0.15	5	7	5	huber
8	20	0.15	5	12	5	huber
9	100	0.05	5	17	5	huber
10	60	0.05	5	17	3	huber
11	100	0.05	5	7	3	huber
12	100	0.2	5	17	5	huber
13	100	0.2	5	7	1	absolute_error
14	100	0.2	5	2	3	absolute_error
15	20	0.15	5	12	5	huber
16	60	0.2	5	17	3	huber
17	20	0.2	5	17	3	huber
18	20	0.2	5	17	3	squared_error
19	20	0.2	5	17	5	huber
20	100	0.2	5	2	5	absolute_error

Com exceção da profundidade máxima, que optou pela mínima profundidade explorada em todas as iterações, todos os outros parâmetros variaram bastante:

- n° de estimadores médio foi de 64
- A taxa de aprendizagem teve média de 0,157
- Amostras por divisão foi 12,75
- Amostras por folha foi de 3,7

2.3.6. Regressão Linear Múltipla

Não explora parâmetros ou mesmo utiliza aprendizagem de máquina, portanto essa etapa não é requerida.

2.4. Avaliação dos modelos

Levando em consideração os hiperparâmetros encontrados durante o treinamento e validação, foram realizados os testes de acordo com a divisão previamente realizada da base de dados e coletados suas respectivas raízes de erro médio quadrático:

Iteração	KNN	SVM	MLP	RF	GB	RLM
1	10.743	10.791	11.082	4.199	4.272	5.821
2	11.623	11.180	11.119	5.222	5.476	6.561
3	11.504	11.036	11.062	5.038	5.069	6.394
4	10.760	10.897	6.048	4.433	4.733	5.753
5	10.769	11.172	11.204	4.353	4.380	5.990
6	11.224	11.562	11.536	4.512	4.590	5.897
7	10.966	10.831	11.104	3.944	3.913	5.592
8	11.805	12.448	11.973	4.429	4.404	6.134
9	11.063	11.318	11.162	5.146	5.203	6.681
10	11.043	11.453	6.788	4.801	5.016	6.413
11	10.598	10.348	10.288	4.320	4.425	5.812
12	11.469	11.452	11.344	4.630	4.656	6.106
13	10.793	11.135	11.037	4.898	5.001	6.112
14	11.828	12.259	8.710	4.422	4.596	6.079
15	10.991	11.284	11.164	5.054	5.022	6.349
16	10.553	10.444	10.531	4.544	4.460	5.780
17	10.415	10.914	10.690	3.893	4.001	5.466
18	10.974	11.002	11.102	4.416	4.541	5.698
19	11.058	11.166	11.067	4.476	4.434	6.103
20	10.896	11.268	11.116	4.760	5.046	6.229
Média (DP)	**10.983 (406,03)**	**11.169 (503,22)**	**11.092 (1538,45)**	**4.494 (370,29)**	**4.593 (404,76)**	**6.091 (326,23)**

Após 20 iterações, pode-se observar que, no geral, os regressores não tiveram bons resultados. Falando de maneira mais aprofundada:

- O SVM obteve o pior desempenho dentre todos, mas o erro foi próximo entre ele, o KNN e o MLP
- Nota-se que o MLP teve o maior desvio padrão por conta de alguns valores muito discrepantes. Fazendo uma análise paralela aos parâmetros escolhidos nessas iterações,

supõe-se que tenha sido algo envolvendo o conjunto de dados já que não houve variação significativa entre os parâmetros. Porém essa questão é observada apenas no MLP

- O Random Forest e o Gradient Boosting foram previsivelmente os melhores métodos - por serem sistemas múltiplos - com uma boa margem
- Por não utilizar aprendizado de máquina, o método de Regressão Linear Múltipla se saiu surpreendentemente bem se comparado aos outros
- Os métodos não conseguiram resultados muito positivos de maneira geral, possivelmente por exploração de parâmetros insuficiente dado o prazo curto
- Os atributos também podem ter sido um problema visto que a correlação entre a maior parte deles não era considerável. Em muitos, ela era quase nula e isso pode ter afetado no desempenho dos métodos

2.5. Análise comparativa

Para a análise, primeiro é realizado o teste de Kruskal-Wallis com 5% de significância como instruído pela descrição da atividade. Foram analisados então os valores apresentados acima e obteve-se como resultado um p-value de 8.495187580123896e-20. Levando em conta os 0.05 de significância, conclui-se que há sim diferença significativa entre os classificadores, logo H_0 é nulo.

Agora realizando os testes de Mann-Whitney bicaudal com 5% de significância entre cada um dos métodos, tem-se que:

	SVM	MLP	RF	GB	RLM
KNN	0,2853047159	0,8817307917	6,80E-08	6,80E-08	6,80E-08
SVM		0,1635957256	6,80E-08	6,80E-08	6,80E-08
MLP			6,80E-08	6,80E-08	3,42E-07
RF				0,5249869102	6,80E-08
GB					7,90E-08

Nota-se por essa comparação que:

- Regressão Linear Múltipla é significativamente distinta de todos os métodos, incluindo Random Forest e Gradient Boosting, que tinham os resultados mais próximos
- Como esperado, SVM, MLP e KNN não tiveram diferença significativa. Inclusive KNN e MLP foram bastante semelhantes
- Random Forest foram muito próximos um do outro, longe de terem diferença significativa, o que era esperado por utilizarem técnicas parecidas de regressão

Com base na comparação, é possível inferir que os melhores métodos de regressão para a base aplicada são Random Forest e Gradient Boosting, que possuem os menores erros e não são significativamente distintos um do outro.