Habitação x valor: modelando o preço médio de casas usando regressão de Ridge

1st Vinicius Costa Secundino
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
viniciussec@hotmail.com

2nd Caio dos Santos Nascimento
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
caio.santos@alu.ufc.br

3rd Paulo Douglas Melo da Silva
Departamento de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
douglascomp@alu.ufc.br

***Resumo*—O presente trabalho busca realizar um estudo sobre a influência de certos fatores, tais como taxa de crime, acessibilidade à rodovias e idade média, sobre o preço médio das habitações na cidade de Boston, localizada em Massachusetts, EUA. Para tal análise, foi criado um modelo de regressão linear simples e a de Ridge (com método de encolhimento) com os conhecimentos adquiridos na disciplina de Inteligência Computacional Aplicada (ICA). Como resultado, as métricas usadas para a análise, coeficiente de determinação (R^2) e erro da raiz da média quadrática (RMSE, em inglês), mostraram resultados abaixo do esperado: $R^2 < 0.75$ e RMSE não reduzindo valor na comparação dos métodos de regressão utilizados.**

***Index Terms*—Predição, regressão linear, método de Ridge, data science ,Boston, imóvel, valor médio**

I. Introdução

Devido ao excesso de dados criados na era da informação, e ao armazenamento desses dados em diversos servidores, surge a oportunidade de manipulá-los e interpretá-los para auxiliar no entendimento de determinadas situações, seja em pesquisas acadêmicas ou aplicações em empresas. No entanto, é preciso ter o domínio de quais e como foram obtidos os dados, além de compreender como e em que se aplicam as técnicas de aprendizado estatístico diante das necessidades dos interessados.

Com base nisso, diante da gama de aplicações do ecossistema da análise de dados, destaca-se a predição de valores de habitações baseada em alguns critérios associados à vizinhança dos locais, pois é de sabedoria comum que esses fatores são de extrema relevância no processo de escolha, por exemplo, casas situadas em regiões com alta frequência de assaltos tendem a valer menos do que casas em locais mais seguros ou condomínios que possuem a própria segurança e as mais próximas aos centros comerciais da cidade tendem a valer mais do que as que estão afastadas. Essas observações aparentemente objetivas variam de acordo com os desejos dos possíveis compradores e, portanto, reitera-se que o escopo da análise de dados é prover resultado o mais próximo possível da realidade diante das informações reunidas.

Este artigo compila o trabalho dos autores na execução da análise descritiva e, posteriormente, preditiva do conjunto de dados das casas de Boston [1] fornecido pela Universidade de Toronto. Como boa prática ao trabalhar com uma grande quantidade de informações é, após breve análise de aspectos mais gerais de cada preditor disponível, executar o pré-processamento dos dados, implicando escolher eliminar ou substituir dados faltantes ou que foram obtidos de forma errada por algum sensor. Com isso, o conjunto de dados agora limpo pode retornar valores mais precisos na futura predição.

O próximo passo quando se deseja fazer uma regressão e vários preditores ainda estão disponíveis no conjunto de dados é analisar a correlação entre os diversos preditores e selecionar os que tiverem maior coeficiente de correlação com a variável que queremos realizar a predição. Além dessa etapa, há como realizar uma Análise de Componentes Principais (PCA) que auxilia também na redução do número de preditores, pois conseguimos fazer uma análise mais precisa, mesmo para uma grande quantidade de dados. A técnica da PCA consiste em calcular componentes que são vetores dos coeficientes de cada preditor em possíveis regressões. Para efetuar esse cálculo, tomando uma combinação linear, cuja forma é a mesma da equação 1, da multiplicação de valores de carga com amostras dos valores dos preditores mais correlatos, busca-se maximizar a expressão da equação 2 para encontrar esses valores normalizados de acordo com a equação 3. Diante do resultado do cálculo são escolhidos os componentes que possuem maior proporção de variância (pois isso indica que aquele componente consegue fazer um ajuste mais correto para a regressão) e é feita uma plotagem que mostra a distribuição dos dados e os vetores associados aos preditores.

$$\mathcal{Z}_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + ... + \phi_{p1}x_{ip} \tag{1}$$

$$\frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{p}\phi_{j1}x_{ij})^2 \tag{2}$$

$$\sum_{j=1}^{p}(\phi)_{ji}^2 = 1 \tag{3}$$

Em relação à disponibilidade das regressões aplicáveis estudadas, havia a normal e as penalizadas de Ridge, Lasso [2] e ElasticNet, todas essas com fácil implementação no software R [3] utilizado no trabalho. Como forma de comparação, a análise das métricas (equações 4 e 5) raiz do erro quadrático médio (RMSE), que calcula a diferença entre o valor real

da informação e o valor da mesma na curva de melhor ajuste, e o coeficiente de determinação R^2, que expressa a quantidade da variância dos dados, usando x_i para representar os dados dos preditores e y_i para representar os dados de saída relacionados, aplicamos o modelo linear ($y = \beta_0 + \beta_1 x$), nas regressões simples, que ajusta os coeficientes visando reduzir a soma quadrática dos erros residuais (RSS), técnica conhecida como abordagem dos mínimos quadrados, com as principais fórmulas representadas nas equações 6, 7 e 8:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{5}$$

$$RSS = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + ... + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n) \tag{6}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}))}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{7}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{8}$$

e de Ridge, esta escolhida pois buscou-se um método para minimizar a expressão mostrada na equação 9 que representa o erro quadrático médio somado a um fator que regula o impacto de parte (primeiro até o j-ésimo) dos coeficientes a serem calculados, ou seja, os primeiros p coeficientes com valores próximos a 0 teriam menos influência na regressão uma vez que seus valores tenderiam a 0.

$$RSS + \lambda \sum_{j=1}^{p}(\beta_j)^2 \tag{9}$$

Diante disso, foi buscado, dentre as regressões feitas, assim como as aplicações dos métodos de embaralhamento, o menor RMSE e o maior R^2.

Outro acessório estatístico é a Regressão de Componentes Principais (PCR), útil para refinarmos ainda mais a quantidade de preditores que realmente têm relevância para a regressão, uma vez que pode haver correlações entre os preditores escolhidos e, portanto, dois ou mais deles poderiam ser ignorados não causando alteração significativa nos resultados da regressão e melhorando a velocidade nas etapas de processamento. Nessa técnica são considerados apenas alguns componentes principais, da mesma gama usada na PCA, para fazer o modelo de regressão linear usando mínimos quadrados. Se esses componentes selecionados conseguirem explicar boa parte da variação dos dados, então o PCR tende a gerar um modelo que consegue generalizar melhor para novas entradas.

Finalmente, foram fixados os conceitos, assim como o uso de diversas ferramentas, estatísticos na aplicação das regressões, penalizadas ou não, o que também era uma objetivo paralelo e igualmente importante na realização deste trabalho.

II. Métodos

O conjunto de dados é formado por atributos relativos às moradias da cidade de Boston, com um total de 506 amostras. Os atributos associados estão listados a seguir:

- CRIM - Taxa de crimes per capita por cidade
- RM - Número de quartos por habitação
- AGE - Idade média da população
- INDUS - Proporção de indústrias presentes
- ZN - proporção de terrenos residenciais zoneados para lotes com mais de 25.000 pés quadrados
- CHAS - Proximidade de um rio (1 se fizer fronteira com rio)
- NOX - Concentração de óxido de nitrogênio
- DIS - Distâncias ponderadas para cinco centros de empregos de Boston
- RAD - Índice de acessibilidade a rodovias
- TAX - Taxa de imposto sobre a propriedade de valor total por $ 10.000
- PTRATIO - Proporção aluno-professor por cidade
- B - $1000(Bk - 0.63)^2$ onde Bk é a proporção de negros por cidade
- LSTAT - Porcentagem da população com baixo nível socioeconômico
- MEDV - Valor mediano de casas ocupadas em $1000

Nosso modelo se propõe a criar uma regressão linear que possa predizer o valor médio de uma casa dado os valores de alguns dos atributos apresentados.

Para evitar que nosso modelo apresente um viés discriminatório, o atributo B foi desconsiderado nas nossas investigações garantindo o caráter ético do modelo.

Para escolher nossos preditores, utilizamos, com maior importância, a matriz de correlação (apesar de termos feito, também, uma análise básica de relação lógica entre variáveis e o preço de casas, baseadas em conhecimento de mundo geral), que mostra, como o nome sugere, a correlação entre as várias colunas do nosso dataset [1]. Essa matriz pode ser facilmente criada com o software R [3]. A matriz gerada a partir do dataset escolhido está representada na Figura 1:

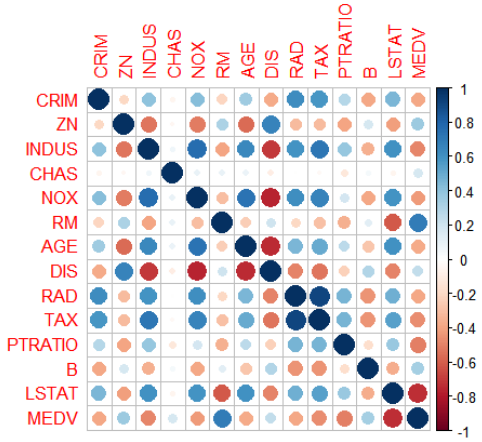


Figura 1: Matriz de correlação

Como gostaríamos de prever a coluna MEDV, nos atentamos para a última coluna da matriz, onde podemos perceber uma maior influência das colunas LSTAT, PTRATIO, INDUS, e RM. Por esse motivo, essas quatro colunas foram escolhidas para serem escolhidas como preditores.

A seguir, realizamos análises mono variadas dos quatro preditores, análise essa que inclui o mapeamento de seus histogramas, cálculo das médias, desvios padrão e obliquidade, utilizando todas as 506 amostras. Tais análises estão representadas na figura 2.

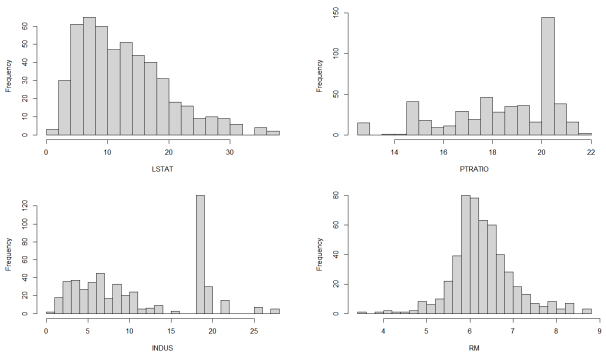


Figura 2: Histogramas

Realizamos cálculos de média, desvio padrão e obliquidade e armazenamos todos os valores em uma mesma tabela, para facilitar visualização. Isso está representado na Tabela I:

	Média	Desvio Padrão	Obliquidade
LSTAT	12.653	7.141	0.901
PTRATIO	18.455	2.164	-0.797
INDUS	11.136	6.860	0.293
RM	6.284	0.702	0.401

Tabela I: Analise monovariada

Podemos destacar o formato gaussiano do gráfico de distribuição do preditor RM, que é responsável pelos dados de quantidade de cômodos por habitação: seu histograma mostra que tanto um número baixo de cômodos como um número alto é incomum em Boston.

Ao realizar a análise bivariada mostrada na Figura 3, onde plotamos os preditores 2 a 2, assim como seus coeficientes de correlação, para podermos ver se há algum nível de correlação entre eles. Assim, vê-se que é fácil ver que os preditores LSTAT e RM tem certa relação linear entre si, mas devido à correlação com a variável alvo foi decidido mantê-las nas regressões.

Para a análise dos componentes principais (PCA), útil para investigar a relação de cada preditor escolhido com a distribuição dos dados no espaço, geramos o gráfico biplot, uma vez que os dados ficam mais visíveis, como é mostrado na Figura 4. Podemos ver na figura alguns elementos: os números representam as amostras trabalhadas, enquanto os vetores (setas) representam os preditores escolhidos.

Pelas características dos preditores, não há necessidade de remoção de outliers, pois todos os dados apresentam uma

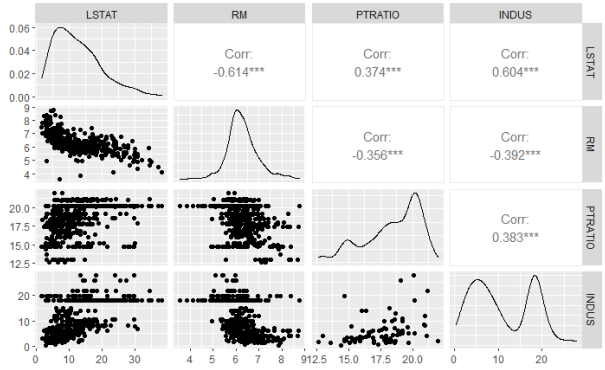


Figura 3: Analise bivariada

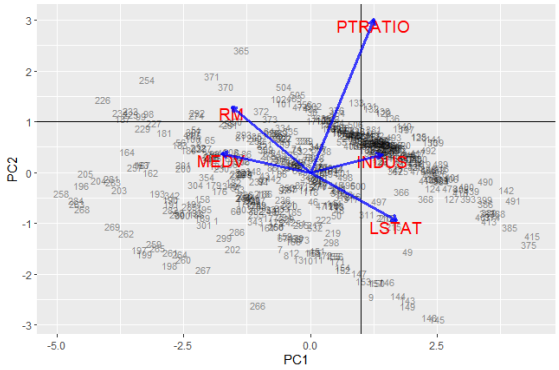
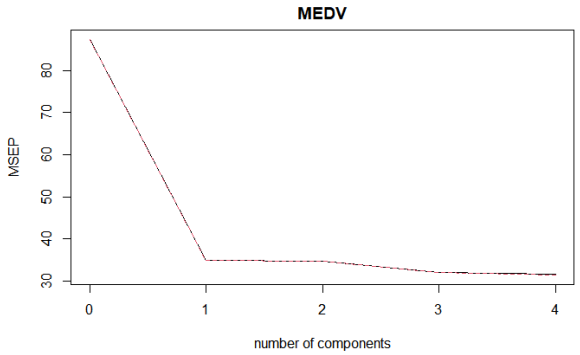


Figura 4: Biplot

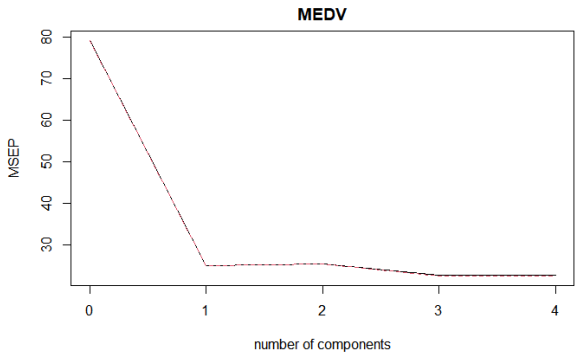
descrição real do conjunto que se forem descartados podem gerar erros na predição dos valores. Pelo observado no gráfico de dispersão (Figura 4), podemos também ver que boa parte dos dados está concentrada na região cujo centro é o par ordenado (1.5, 0.5) da PCA e que não temos sobreposição de nenhum preditor.

Em virtude de justificar a escolha da regressão penalizada de Ridge, buscamos uma que não removesse a influência dos preditores durante o treino, por mais que os coeficientes relacionados fossem baixos. Para retornar o valor ótimo do parâmetro lambda (λ) realizamos algumas iterações buscando o menor valor para possível e obtemos o valor 0.1.

Por fim, a escolha da Principal Component Regression (PCR) em relação ao Partial Least Squares (PLS) é devido ao fato de que, mesmo se as direções dos componentes principais não mostrarem a maior proporção de variação, isso vai indicar uma aproximação para obter bons resultados, além de que o PCR performa melhor com preditores que não se sobrepõem [4] (como vimos no plot do PCA). Na Figura 5, a execução do código no software R nos mostra que um único componente está impactando cerca de 50% na proporção da variação do erro no primeiro gráfico, (a)- Treino, e esse erro é reduzido ainda mais no segundo gráfico, (b) - teste, mas, para evitar simplificação da regressão, decidimos manter os demais preditores nos plots do PCR:



(a) Treino



(b) Teste

Figura 5: Variação do erro médio quadrático da predição (MSEP) calculado pela PCR

III. Resultados

Estando com os preditores escolhidos, realizamos uma separação do conjunto de dados entre dados de treino e dados de teste. Foram escolhidos 70% dos dados para treinamento do modelo de regressão e os demais 30% para teste.

O modelo de regressão foi criado utilizando a regressão linear ordinária [2] e otimizado pela validação cruzada de 10 campos. Este tipo de validação consiste em separar os dados disponíveis em 10 subconjuntos aproximadamente com o mesmo tamanho e realizar por dez iterações o processo de escolher um desses subconjuntos para ser o de validação enquanto que os demais servem para treino e são calculadas as métricas. Ao final desse processo é tomada a média das métricas, considerando todas as iterações, como resultado global.

Realizamos a análise das métricas das regressões nos conjuntos de treino e teste em cada modelo e colocamos na Tabela II:

	RMSE	R-squared
Linear	5.458	0.657
Linear (10-foldCV)-teste	5.477	0.654
Ridge	5.470	0.655
Ridge-teste	4.605	0.728
Ridge (10-foldCV)	5.474	0.655
Ridge (10-foldCV)-teste	4.582	0.731
PCR-treino	11.375	-0.489
PCR-teste	9.469	-0.145

Tabela II: Regressões

Dados os resultados apresentados, destacamos alguns valores da métrica R^2 negativos, o que indica que o modelo de regressão associado (PCR) teve performance pior do que se fosse considerado um modelo de regressão linear horizontal. Além disso, comparando o modelo linear comum com a regressão de Ridge, houve um aumento significativo na métrica R^2, o que indica que o modelo penalizado foi ajustado de forma a conseguir melhor generalização para novos dados, mas não houve muita variação comparando as métricas ao treinar as regressões de Ridge de forma direta (separando o conjunto de dados em um grupo de treino e outro de teste apenas uma vez) e ao utilizar Ridge com validação cruzada (CV, Cross Validation) de dez campos o que indica que o conjunto de dados possui boa separabilidade, ou seja, tomando ou não amostras diferentes para realizar o ajuste, conseguimos modelos com capacidade semelhante para generalizar adequadamente para novas entradas. Apesar disso, de forma geral, o resultado das métricas (alto RMSE e baixo coeficiente de determinação) indicam um modelo de regressão que não se ajustou adequadamente, ou seja, o resultado obtido não foi o esperado.

Concluindo, podemos afirmar que a execução do desenvolvimento dos modelos pôde ser bastante engrandecedora pois todos os conceitos foram revisados em livros e sites diversos e aplicados via software, auxiliando na fixação do conhecimento aprendido durante esta etapa do semestre.

Referências

[1] Harrison, D. and Rubinfeld, D.L. 'Hedonic prices e the demand for clean air', J. Environ. Economics & Management, vol.5, 81-102, 1978.

[2] G. James, D. Witten, T. Hastie e R. Tibshirani. "An Introduction to Statistical Learning with Applications in R"2013, Springer.

[3] R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[4] Scikit Learn. Principal Component Regression vs Partial Least Squares Regression URL https://scikit-learn.org/stable/auto_examples/cross_decomposition/plot_pcr_vs_pls.html

LISTA DE CÓDIGOS

Importação de dados e análise monovariada

```
d1=read.csv("Boston.csv",sep=",",header=TRUE)

cols = names(d1)

for (i in cols) {
  hist(d1[i])
}

df <- data.frame( LSTAT = d1$LSTAT, RM= d1$RM,
    PTRATIO = d1$PTRATIO, INDUS = d1$INDUS, MEDV =
    d1$MEDV)

media <- sapply(Filter(is.numeric, df[, 2:5]), mean)

hist(d1$LSTAT)
desvpad <- sapply(Filter(is.numeric, df[, 2:5]), sd)

#install.packages("timeSeries")
library(timeSeries)

obliq <- colSkewness(df[, 2:5],pvalue=true)

resumo <- data.frame(media=media, desvioPadrao=
    desvpad, skewness = obliq)
resumo
```

Plotagem dos preditores 2 a 2

```
library("ggplot2")
library("GGally")

ggpairs(df[, 1:4])
```

Matriz de correlação visual

```
#install.packages("corrplot")

library(corrplot)

res <- cor(d1)
round(res, 2)

corrplot(res, method="circle")
```

PCA

```
#install.packages("ggfortify")
library(ggfortify)
df <- data.frame( LSTAT = d1$LSTAT, RM= d1$RM,
    PTRATIO = d1$PTRATIO, INDUS = d1$INDUS, MEDV =
    d1$MEDV)

PCbiplot <- function(PC, x="PC1", y="PC2", colors=c(
    'black', 'black', 'red', 'red')) {
    # PC being a prcomp object
    #data <- data.frame(obsnames=row.names(PC$x), PC
    $x)
  data<-data.frame(obsnames= 1:506, PC$x)
    plot <- ggplot(data, aes_string(x=x, y=y)) +
    geom_text(alpha=.4, size=3, aes(label=obsnames),
     color=colors[1])

    plot <- plot + geom_hline(aes(0), size=.2,
    yintercept=1, xintercept=1) + geom_vline(aes(0),
     size=.2, color=colors[2], xintercept=1)
    datapc <- data.frame(varnames=rownames(PC$
    rotation), PC$rotation)
    mult <- min(
        (max(data[,y]) - min(data[,y])/(max(datapc[,
    y])-min(datapc[,y]))),
        (max(data[,x]) - min(data[,x])/(max(datapc[,
    x])-min(datapc[,x])))
        )
    datapc <- transform(datapc,
            v1 = .7 * mult * (get(x)),
            v2 = .7 * mult * (get(y))
            )
    plot <- plot + coord_equal() + geom_text(data=
    datapc, aes(x=v1, y=v2, label=varnames), size =
    5, vjust=1, color=colors[3])
    plot <- plot + geom_segment(data=datapc, aes(x
    =0, y=0, xend=v1, yend=v2), arrow=arrow(length=
    unit(0.2,"cm")), alpha=0.75, color=colors[4])
    plot
}

pca.eg <- prcomp(df, scale=T)
PCbiplot(pca.eg, colors=c("black", "black", "red", "
    yellow"))
```

Regressões e cálculo das métricas

```
#install.packages("glmnet")
library(glmnet)
library(caret)



cols = c('LSTAT', 'RM', 'PTRATIO', 'INDUS')

pre_proc_val <- preProcess(train.data[,cols], method
     = c("center", "scale"))

train.data[,cols] = predict(pre_proc_val, train.data
    [,cols])
test.data[,cols] = predict(pre_proc_val, test.data[,
    cols])


lr <- lm(MEDV ~., data = train.data)



eval_results <- function(true, predicted, df) {
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  R_square <- 1 - SSE / SST
  RMSE = sqrt(SSE/nrow(df))


  # Model performance metrics
data.frame(
  RMSE = RMSE,
  Rsquare = R_square
)

}

# Prediction and evaluation on train data
pred_lr <- predict(lr, newx = train.data)
eval_results(train.data$MEDV, pred_lr, train.data)


# Define training control
set.seed(123)
train.control <- trainControl(method = "cv", number
    = 10)
# Train the model
lr_10fold <- train(MEDV ~., data = train.data,
    method = "lm",
```

```
                trControl = train.control)
# Summarize the results
print(lr_10fold)
```

```
cols_reg = c('LSTAT', 'RM', 'PTRATIO','INDUS', 'MEDV
    ')

library("caret")

lett <- dummyVars(MEDV ~ ., data = df[,cols_reg])

train_lett = predict(lett, newdata = train.data[,
    cols_reg])

test_lett = predict(lett, newdata = test.data[,cols_
    reg])

print( dim(train_lett) ); print( dim(test_lett) )


x = as.matrix(train_lett)
y_train = train.data$MEDV


x_test = as.matrix(test_lett)
y_test = test.data$MEDV



##Ridge normal
ridge_reg = glmnet(x, y_train, nlambda = 25, alpha =
     0, family = 'gaussian')

lambda_ord = min(ridge_reg$lambda)

##Ridge com cross validation
set.seed(34)
cv_ridge <- cv.glmnet(x, y_train, alpha = 0, lambda
    = lambdas)
plot(cv_ridge)
optimal_lambda <- cv_ridge$lambda.min
optimal_lambda
```

```
# Predicao e avaliacao nos dados de treino para
    ridge sem validacao cruzada
predictions_train <- predict(ridge_reg, s = lambda_
    ord, newx = x)
eval_results(y_train, predictions_train, train.data)

# Predicao e avaliacao nos dados de teste para ridge
     sem validacao cruzada
predictions_test <- predict(ridge_reg, s = lambda_
    ord, newx = x_test)
eval_results(y_test, predictions_test, test.data)


# Predicao e avaliacao nos dados de treino para
    ridge com validacao cruzada
predic_train <- predict(cv_ridge, s = optimal_lambda
    , newx = x)
eval_results(y_train, predic_train, train.data)

# Predicao e avaliacao nos dados de teste para ridge
     com validacao cruzada
predic_test <- predict(cv_ridge, s = optimal_lambda,
     newx = x_test)
eval_results(y_test, predic_test, test.data)
```

PCR

```
set.seed(2)
```

```
pcr.fit1=pcr(MEDV~., data=test.data,scale=TRUE,
    validation="CV")
summary(pcr.fit1)
validationplot(pcr.fit1,val.type="MSEP")


predic_test_pcr <- predict(pcr.fit1, newx = x_test)
eval_results(y_test, predic_test_pcr, test.data)


set.seed(2)

pcr.fit2=pcr(MEDV~., data=train.data, scale=TRUE,
    validation="CV")
validationplot(pcr.fit,val.type="MSEP")

predic_train_pcr <- predict(pcr.fit2, newx = x)
eval_results(y_train, predic_train_pcr, train.data)

summary(pcr.fit2)
```