Trabalho 1 - Machine Learning (EEL891/EEL860)

Paulo Oliveira Lenzi Valente
DRE:114190931

Kaggle: kaggle.com/polvalente

DEL/Poli/UFRJ

Departamento de Engenharia Eletrônica e de Computação
Escola Politécnica – Universidade Federal do Rio de Janeiro

Professor: Heraldo L. S. Almeida

Rio de Janeiro, 17 de Julho de 2018

Conteúdo

1 Objetivos

Este trabalho visa a criar um regressor linear para a competição House Prices - Advanced Regression Techniques do Kaggle.

1.1 Obtenção dos Dados

Os dados foram obtidos através do sublink: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

2 Pré-Processamento dos Dados

Junto com o *dataset*, veio um aquivo chamado *data_description.txt*. Nele, há uma descrição detalhada sobre cada uma das colunas do dataset. Assim, a partir dele foram selecionadas algumas colunas para se abandonar a priori (que julguei serem subjetivas demais). Além disso, foram enumeradas quais colunas são ordinais e quais são categóricas. A seguir, será descrito o que foi feito com cada uma destas categorias.

2.1 Colunas Removidas

As seguintes colunas foram removidas por representarem categorias, na minha opinião, subjetivas, no sentido de que representariam gosto pessoal do comprador:

- LotShape
- RoofStyle
- RoofMatl
- MasVnrArea
- MasVnrType
- BsmtFinType1
- BsmtFinType2
- GarageFinish
- GarageQual
- GarageCond
- MiscFeature

As próximas são colunas objetivas, mas que julguei serem descartáveis, dado que não temos tantos dados e seria interessante descartar o máximo de colunas o possível. Foram descartadas apenas do sexto experimento em diante.

- LotFrontage
- PavedDrive
- GarageArea (esta está ainda representada no número de carros)
- WoodDeckSF
- OpenPorchSF
- EnclosedPorch
- 3SsnPorch
- ScreenPorch

2.2 Colunas Ordinais

- OverallQual
- OverallCond
- ExterQual
- ExterCond
- BsmtQual
- BsmtCond
- BsmtExposure
- KitchenQual
- FireplaceQu
- PoolQC
- Fence
- SaleCondition
- LandSlope
- MoSold

2.3 Colunas Categóricas

- MSSubClass
- MSZoning
- Street
- Alley
- LandContour
- Utilities
- LotConfig
- Neighborhood
- Condition1
- Condition2
- BldgType
- HouseStyle
- Exterior1st
- Exterior2nd
- Foundation
- Heating
- HeatingQC
- CentralAir
- Electrical
- Functional
- GarageType
- SaleType

2.4 Tratamento dos datasets

Ao tentar carregar os dados de teste, separadamente dos dados de treinamento, houve erros devido a "novas categorias". Inspeção dos arquivos .csv indicaram que houve preenchimento errado dos dados. Isso foi corrigido através de comandos "find and replace" no Visual Studio Code.

Para o último experimento, a coluna de *SalePrice*, a variável independente, foi tratada através de um filtro $log(y + 1)$, para compensar sua kurtose.

3 Experimentos e Resultados

Os parágrafos a seguir indicarão a evolução temporal dos experimentos e os desempenhos subsequentes.

Para o primeiro experimento, como linha de base, escolhi treinar o regressor SVM padrão. Não foi passado nenhum parâmetro extra. O resultado obtido no Kaggle foi de 0.41885.

Depois, para experimentar com escala, mudei para o StandardScaler + SVM. Com isso, a pontuação obtida foi de 0.41872, o que é uma melhoria, mas não é uma diferença significativa.

A terceira tentativa foi com o RidgeCV + StandardScaler, utilizando *alphas* de 0.01, 0.05, 0.1, 0.5, 1, 5, 10 e 50. Nesse, a pontuação obtida foi de 0.44956. Com isso, resolvi voltar ao SVM.

A quarta tentativa utilizou *grid search* com *cross-validation* sobre um regressor SVM. No *grid search* se utilizou os parâmetros:

- C: 1, 10, 100, 1000
- kernel: linear, rbf

O melhor estimador para o experimento foi com C = 100 e kernel = linear. A pontuação foi de 0.17142, uma melhora considerável, que me fez subir 1605 posições no ranking. Isso me levou a continuar com SVM, experimentando com uma malha de busca mais complexa.

O quinto experimento consistiu nos seguintes parâmetros para o Grid Search:

- C: 1, 5, 10, 50, 100, 500, 1000
- epsilon: 0.001, 0.01, 0.1, 1
- kernel: linear, rbf, poly
- degree: 2, 3, 4

O melhor estimador foi com os parâmetros: C = 50.0, degree = 2, epsilon=1.0, kernel='linear'. Pontuação: 0.16745

No sexto experimento, foram removidas mais *features* (indicadas na seção de pré-processamento). O grid do quinto experimento foi repetido. Isso levou a uma pontuação de 0.15970, com os parâmetros: C = 50, degree = 2, epsilon = 1, kernel = linear.

No sétimo experimento, se reduziu a resolução de 'Month Sold' para trimestres e simplificou a malha de busca para:

- C: 1, 5, 10, 50, 100, 500, 1000
- epsilon: 0.001, 0.01, 0.1
- kernel: linear
- degree: 2, 3, 4, 5

O melhor estimador, com C = 50, degree = 2, epsilon = 0.1 e kernel = linear, levou a uma pontuação de 0.15945, uma piora leve nos resultados.

Alguns outros experimentos supérfluos foram realizados. Neles, não houve melhora dos resultados. Se utilizou regressão Lasso.

O experimento final, com a melhor pontuação, se deu da seguinte maneira:

No pré-processamento, apliquei log(1 + y) na coluna de SalePrices, e apliquei a operação inversa para calcular o preço estimado após a predição. Foi utilizado um grid mais simples, pois constatei que sempre era selecionado o SVM linear de grau 2:

- C: 1, 5, 10, 50, 100
- gamma: 0.001, 0.01, 0.1
- kernel: linear
- degree: 2

O estimador selecionado foi: C = 100, degree = 2, gamma = 0.001, kernel = linear Pontuação: 0.14881