Predictive modeling for house prices in Boston

Lee Anna 18th Dec 2016

Source: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

How do home features add up to its price tag?

This project challenges to predict the final price of each home.

For each Id in the test set, the value of the SalePrice variable would be predicted.

a) 데이터에 대한 이해/생각

1. Train set와 Test set으로 나눠져 있음
2. 각각 1500 이하의 Rows가 있으며 Columns 개수 또한 80개(Without SalePrice)로 많은 편
3. Train 데이터에 Integer(35), float(3), string(43) 이 섞여있음
4. NaN 값이 많아 정리를 해야 한다. 삭제 또는 Mean 값을 넣을 수도 있는데 rows 삭제대신 Mean 값을 넣기로 결정
5. Qualitative 데이터가 quantitative 데이터보다 조금 더 많음. Regression 위주로 분석하기 위해서 Categorical features들은 분석 시 제거
6. Test set은 SalePrice column이 없다. 따라서 Response를 세팅할 수 없으므로 Train 데이터로 모델을 만든 후 Test set을 사용하여 돌리면 됨

b) 왜 이 목표를 잡았는지

1. 개인적으로 부동산에 관심이 많고, 프로젝트를 할 때 흥미 있는 분야를 선택해야 데이터에 대한 이해도가 더 높을 것
2. Classification보다는 Regression 모델들을 만들고 싶었고 features가 많으니 여러 방향으로 분석이 용이
3. Predictors들에 대한 자세한 설명이 Kaggle 웹 페이지에 나와있어 시간 절약 가능
4. Kaggle에 올라온 kernels와 비교해보면서 내가 만든 project의 완성도/수준 확인 가능; 배우지 않은 기법들 e.g. XGBoost을 사용했기 때문에 이 프로젝트 분석에 참고 하지 않았음

c) 왜 이 방법을 사용했는지

1. 크게 Linear regression과 Tree based methods로 사용함

- Linear regression: simple linear regression, multiple linear regression, interaction terms, scikit-learn을 이용한 multiple linear regression
- Tree based methods: Regression Tree, Random Forest, Gradient Boosting, train/test split approach, K-Fold Cross Validation

2. Linear regression은 간단하고 코드량도 적게 들어가므로 빠른 시간 내에 여러 모델을 만들어 보면서 테스트가 가능
3. Cross validation은 tree 모델들에 적당한 사이즈나 정해주기 위한 방법으로 사용
4. Test와 Train set이 이미 나눠져 있으나 시간상 Train set에서 Train/Test split validation set approach를 사용함. 교수님이 주신 자료를 참고하여 적용

d) 결과가 왜 이런지

1) Prediction model1: **Simple Linear Regression**

- 집 면적이 Sale Price에 가장 영향을 줄 수 있다고 생각하여 GrLivArea를 선택
- 결과: SalePrice와 GrLivArea는 서로 관련이 있고 모델은 유효하다.

2) Prediction model2: **Multiple Linear Regression**

- GrLivArea 뿐만 아니라 Overall Quality, Total Basement Square Feet 그리고 Garage에 들어가는 car 수를 선택함으로써 더 많은 조건이 더 나은 예측할 수 있도록 함
- 좋은 predictors들만 사용함으로써 예측 능력이 좋았음

3) Prediction model3: **Interaction effect**

- 먼저 Overall Quality와 집이 지어진 연도의 상관관계가 있는지 증명. P-value가 0.000으로 나왔으므로 서로 관련이 있음
- 두 features 사이에 시너지 효과로 SalePrice 예측 가능. 심지어 GrLivArea만 사용한 model 1보다 예측을 잘함

4) Prediction model4: **scikit-learn Linear Regression**

- Machine Learning algorithm을 사용하는 scikit-learn 사용
- Model 2에서 선택한 feature 4개로 X에 Data frame 지정.
- y는 Response이므로 SalePrice 지정
- 해당 X와 y를 이용하여 model을 훈련시킴. 조건이 같으니 model 2의 output과 같음
- 새로운 feature 값을 주고 response 추정해봄

5) Prediction model5: **Regression Tree**

- Feature 2개를 지정해서 tree를 만드는 것은 쉽지만 모든 features들을 포함 시 Qualitative columns과 NaN값 때문에 에러 발생. get_dummies를 사용하여 categorical columns은 제거하고 전체에 퍼져있는 NaN값은 mean 값을 넣음

- K-Fold Cross Validation을 사용해 best tree size 측정하여 max_leaf_nodes를 9으로 잡고 새로 트리 생성

6) Prediction model6: **Random Forest**

- Cross Validation으로 best feature size 측정하여 8이 나옴
- Train/Test split 방식으로 나누고 Variable Importance를 그래프로 보니 주요 predictors들을 쉽게 파악할 수 있었음
- Training set으로 예측 시 정확하게 다 맞추었으나 test set은 오차가 조금 있음

7) Prediction model7: **Gradient Boosting**

- 여기도 Train/Test split을 사용
- Tree 개수를 점점 늘려가며 성능을 보다가 Cross Validation으로 측정하여 Best number of tree는 1100인 것으로 확정
- n_estimators를 1100으로 맞추고 예측 결과를 보니 random forest보다 정확하다는 것을 확인할 수 있었음

8) Conclusion

- Predictors들을 다 쓴다고 좋은 것은 아니다. 필요한 것만 골라 적용하는 것이 오히려 이득
- Variable importance를 먼저 측정하는 것이 좋겠음. Linear regression 모델 만들 때 어떤 predictor가 좋을지 고민하지 않아도 됨

9) Result

- Test.csv를 로드한 후 Training set을 사용하여 만들어 놓았던 lm_mul와 GRB_reg_x 모델을 사용하여 SalePrice prediction 결과 출력

e) 더 해보고 싶지만 하지 못한 것

- Classification 모델을 만들어 볼 수 있는 feature들도 e.g. Quality-good/bad 여러 개 있었지만 SalePrice의 정확한 측정을 이끌어 내는데 많은 도움 줄지 의문이었고, 이 프로젝트의 목표는 애초부터 regression 이었기 때문에 분석에서 제외하였습니다.
- 충분한 시간이 있다면 Neighborhood, Proximity to main road, House Style 등의 Sale Price에 대한 영향력 분석을 하고 싶습니다.