Report of Applied Machine Learning Project

House Prices: Advanced Regression Techniques

Team Members:
Burak Can Onarım
Ramazan Ayöz
Salih Gireniz

1. GOAL / MOTIVATION

This dataset will help us to find out an ideal price of a house which we want to buy and it will help us to compare that house with other houses. So, we can focus on the price that seller told us and we can question that ‘Does it worth to pay?’

2. DESCRIPTION OF DATASET

Our data has 81 attibutes with

- 1 Primary Key
    1. ID

- 34 Integers
    1. MSSubClass
    2. LotArea
    3. OverallQual
    4. OverallCond
    5. YearBuilt
    6. YearRemodAdd
    7. BsmtFinSF1
    8. BsmtFinSF2
    9. BsmtUnfSF
    10. TotalBsmtSF
    11. 1stFlrSF
    12. 2ndFlrSF
    13. LowQualFinSF
    14. GrLivArea
    15. BsmtFullBath
    16. BsmtHalfBath
    17. FullBath
    18. HalfBath
    19. BedroomAbvGr

20. KitchenAbvGr
21. TotRmsAbvGrd
22. Fireplaces
23. GarageCars
24. GarageArea
25. WoodDeckSF
26. OpenPorchSF
27. EnclosedPorch
28. 3SsnPorch
29. ScreenPorch
30. PoolArea
31. MiscVal
32. MoSold
33. YrSold
34. SalePrice

- 46 Strings
    1. MSZoning
    2. LotFrontage
    3. Street
    4. Alley
    5. LotShape
    6. LandContour
    7. Utilities
    8. LotConfig
    9. LandSlope
    10. Neighborhood
    11. Condition1
    12. Condition2
    13. BldgType
    14. HouseStyle
    15. RoofStyle
    16. RoofMatl
    17. Exterior1st
    18. Exterior2nd
    19. MasVnrType
    20. MasVnrArea
    21. ExterQual
    22. ExterCond
    23. Foundation
    24. BsmtQual

25. BsmtCond
26. BsmtExposure
27. BsmtFinType1
28. BsmtFinType2
29. Heating
30. HeatingQC
31. CentralAir
32. Electrical
33. KitchenQual
34. Functional
35. FireplaceQu
36. GarageType
37. GarageYrBlt
38. GarageFinish
39. GarageQual
40. GarageCond
41. PavedDrive
42. PoolQC
43. Fence
44. MiscFeature
45. SaleType
46. SaleCondition

Also, we have 1460 rows in train set and 1459 rows in test set. Finally, this link will giving the information about our data: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data?

3. THINGS WE DID

1. In the preprocessing, we've changed attribute names and removed current year:

```
names(train)[5]
names(train)[names(train) == "YearBuilt"] <- "Age"
names(test)[names(test) == "YearBuilt"] <- "Age"
train$Age <- 2019 - train$Age
```

2. Then, we converted to numeric values, because they are ordinal and meaningful for numbers:

```
train$PoolQC <- as.numeric(factor(train$PoolQC,
                                  levels = c("Ex", "Fa","Gd", "TA","Po"),
                                  labels = c(5,2,4,3,1) ,ordered = TRUE))

train$ExterQual <- as.numeric(factor(train$ExterQual,
                                     levels = c("Ex", "Fa","Gd", "TA","Po"),
                                     labels = c(5,2,4,3,1) ,ordered = TRUE))

train$ExterCond <- as.numeric(factor(train$ExterCond,
                                     levels = c("Ex", "Fa","Gd", "TA","Po"),
                                     labels = c(5,2,4,3,1) ,ordered = TRUE))


train$GarageCond <- as.numeric(factor(train$GarageCond,
                                      levels = c("Ex", "Fa","Gd", "TA","Po"),
                                      labels = c(5,2,4,3,1) ,ordered = TRUE))

train$GarageQual <- as.numeric(factor(train$GarageQual,
                                      levels = c("Ex", "Fa","Gd", "TA","Po"),
                                      labels = c(5,2,4,3,1) ,ordered = TRUE))


train$BsmtQual <- as.numeric(factor(train$BsmtQual,
                                    levels = c("Ex", "Fa","Gd", "TA","Po"),
                                    labels = c(5,2,4,3,1) ,ordered = TRUE))

train$BsmtCond <- as.numeric(factor(train$BsmtCond,
                                    levels = c("Ex", "Fa","Gd", "TA","Po"),
                                    labels = c(5,2,4,3,1) ,ordered = TRUE))


train$HeatingQC <- as.numeric(factor(train$HeatingQC,
                                     levels = c("Ex", "Fa","Gd", "TA","Po"),
                                     labels = c(5,2,4,3,1) ,ordered = TRUE))

train$KitchenQual <- as.numeric(factor(train$KitchenQual,
                                       levels = c("Ex", "Fa","Gd", "TA","Po"),
                                       labels = c(5,2,4,3,1) ,ordered = TRUE))

train$FireplaceQu <- as.numeric(factor(train$FireplaceQu,
                                       levels = c("Ex", "Fa","Gd", "TA","Po"),
                                       labels = c(5,2,4,3,1) ,ordered = TRUE))
```

3. And then, we handled the missing data. For example, if there is no garage, it is NA but NA is not missing, actually:

GarageCond	GarageCond	81
BsmtExposure	BsmtExposure	38
BsmtFinType2	BsmtFinType2	38
BsmtQual	BsmtQual	37
BsmtCond	BsmtCond	37
BsmtFinType1	BsmtFinType1	37
MasVnrType	MasVnrType	8
MasVnrArea	MasVnrArea	8
Electrical	Electrical	1
Id	Id	0
MSSubClass	MSSubClass	0
MSZoning	MSZoning	0
LotArea	LotArea	0

4. Now, we can handle with missing data like that:

```
#3#fireplace yoksa NA koyulmuşn düzeltilmesi gerek bunuMissing_Values > 0,]
#some NA entries in the test sets actually mean "no Garage"
train$Alley[is.na(train$Alley)] = "No alley access"
train$BsmtQual[is.na(train$BsmtQual)] = 0
train$BsmtCond[is.na(train$BsmtCond)] = 0
train$BsmtExposure[is.na(train$BsmtExposure)] = "No basement"
train$BsmtFinType1[is.na(train$BsmtFinType1)] = "No basement"
train$BsmtFinType2[is.na(train$BsmtFinType2)] = "No basement"
train$FireplaceQu[is.na(train$FireplaceQu)] = 0
train$GarageType[is.na(train$GarageType)] = "No garage"
train$GarageFinish[is.na(train$GarageFinish)] = "No garage"
train$GarageQual[is.na(train$GarageQual)] = 0
train$GarageCond[is.na(train$GarageCond)] = 0
train$PoolQC[is.na(train$PoolQC)] = 0
train$Fence[is.na(train$Fence)] = "No fence"
train$MiscFeature[is.na(train$MiscFeature)] = "None"
train$MasVnrType[is.na(train$MasVnrType)] = "None"
train$Electrical[is.na(train$Electrical)] = "SBrkr"
train$LotFrontage[is.na(train$LotFrontage)] = median(train$LotFrontage, na.rm = TRUE)
#we use -9999 numeric because non-sensical value
train$MasVnrArea[is.na(train$MasVnrArea)] = -9999
train$MasVnrArea
train$GarageYrBlt[is.na(train$GarageYrBlt)] = -9999
train$GarageYrBlt
```

5. After the missing values, we converted them factor(nominal), because they are characters:

```
###Factorizing
train$MSZoning<- factor(train$MSZoning)
train$Street <- factor(train$Street)
train$LotShape <-factor(train$LotShape )
train$LandContour<-factor(train$LandContour)
train$Utilities<-factor(train$Utilities)
train$LotConfig<-factor(train$LotConfig)
train$LandSlope<-factor(train$LandSlope)
train$Neighborhood<-factor(train$Neighborhood)
train$Condition1<-factor(train$Condition1)
train$Condition2<-factor(train$Condition2)
train$BldgType<-factor(train$BldgType)
train$HouseStyle<-factor(train$HouseStyle)
train$RoofStyle<-factor(train$RoofStyle)
train$RoofMatl<-factor(train$RoofMatl)
train$Exterior1st<-factor(train$Exterior1st)
train$Exterior2nd<-factor(train$Exterior2nd)
train$MasVnrType<-factor(train$Exterior2nd)
train$Foundation<-factor(train$Foundation)
train$Heating<-factor(train$Heating)
train$CentralAir<-factor(train$CentralAir)
train$Functional<-factor(train$Functional)
train$PavedDrive<-factor(train$PavedDrive)
train$SaleType<-factor(train$SaleType)
train$SaleCondition<-factor(train$SaleCondition)
train$MiscFeature<-factor(train$MiscFeature)
train$Fence<-factor(train$Fence)
train$GarageFinish<-factor(train$GarageFinish)
train$GarageType<-factor(train$GarageType)
train$BsmtFinType1<-factor(train$BsmtFinType1)
train$BsmtFinType2<-factor(train$BsmtFinType2)
train$BsmtExposure<-factor(train$BsmtExposure)
train$Alley<-factor(train$Alley)
train$Electrical<-factor(train$Electrical)
```

6. Eventually, we can use the file with WEKA after that we converted to ARFF format.

7. We find outlier values for better result and we remove them
Filter >> InterquartileRange
Filter >> RemoveWithValues >> parametreOutliers

8. We select some attributes using CorrelationAttibuteEval and ClassifierAttributeEval

Select attributes >> CorrelationAttibuteEval and ClassifierAttributeEval
Select attributes >> CorrelationAttibuteEval >> and Search method >> Ranker
Select attributes >> ClassifierAttributeEval >> and Search method >> Ranker

And then, we have these attributes:

- LotFrontage
- Neighborhood
- OverallQual
- YearRemodAdd
- ExterQual
- BsmtFinSF1
- TotalBsmtSF
- HeatingQC
- X1stFlrSF
- GrLivArea
- KitchenQual
- TotRmsAbvGrd
- Fireplaces
- GarageFinish
- GarageCars
- GarageArea
- OpenPorchSF
- SalePrice

9. Finally, we've discretize like that:

Discretize Filter >> Choose >> Discretize >> parameters (bins = 10 attribute = 15(Carage Cars))

4. CHALLENGES

We are slog on selecting attributes for finding the SalePrice when we said in 3.8. The hard part is how to decide the attributes and how to be selected these? And then we learn that CorrelationAttibuteEval and ClassifierAttributeEval and we use them. Then, this problem would be solved.

5. RESULTS

Best 4 Models Results with 10 Cross-Validation Folds:

5.1 Linear Regression

Correlation coefficient	0.9381
Mean absolute error	16917.3329
Root mean squared error	24329.0099
Relative absolute error	31.9515 %
Root relative squared error	34.6149 %
Total Number of Instances	1383

5.2 Gaussian Regression

Correlation coefficient	0.9379
Mean absolute error	16952.7493
Root mean squared error	24365.8099
Relative absolute error	32.0184 %
Root relative squared error	34.6673 %
Total Number of Instances	1383

5.3 Tree M5P

Correlation coefficient	0.9443
Mean absolute error	15634.4102
Root mean squared error	23125.4488
Relative absolute error	29.5284 %
Root relative squared error	32.9025 %
Total Number of Instances	1383

5.4 Lazy.LWL
weightingKernel: 2
classifier: Linear Regression

Correlation coefficient	0.9484
Mean absolute error	15432.3832
Root mean squared error	22294.5873
Relative absolute error	29.1469 %
Root relative squared error	31.7204 %
Total Number of Instances	1383