Assignment 2: Regression Model Building

Sri Seshadri

7/1/2017

1. Introduction

This report discusses the regression models for estimating or predicting the sales price of "typical" homes in Ames, Iowa.

2. Sample definition

It is assumed that typical home buyers are those that move from apartments to single family or town homes. Also apartments are less likely to be sold to individuals as they remain holdings of owners for rental income. Single family and town homes belong to "Residential Low density" (RL) zoning classification in the city of Ames. Data belonging to only to the RL zone is considered for analysis and model development. Also, it is assumed that typical homes have paved streets for access and above grade living area greater than 800 square feet. Sales data belonging to homes that were sold in abnormal conditions such as trade in, foreclosure or short sale are not included in the analysis. Also, sales between family members, sale of adjoining lot, linked properties are omitted from the data. Table 1 shows the waterfall of the data not included in the data and the eligible samples.

Table 1: Drop waterfall

DropCondition	counts
01: Not LowDensityZone	657
02: Not Normal/Partial Sale	189
03: Street Not Paved	3
04: Less than 800 SqFt	41
99: Eligible Sample	2040

The following variables in the data were deemed to be of interest for model building. The choice of parameters was based upon intial Exploratory Data Analysis (EDA) and subject matter expertise. See appendix A.1 for data quality checks.

Table 2: Variables of interest

LotArea	TotalBsmtSF	KitchenQual
LotConfig	GrLivArea	TotRmsAbvGrd
Neighborhood	BsmtFullBath	GarageArea
BldgType	BsmtHalfBath	MoSold
HouseStyle	FullBath	YrSold
OverallCond	HalfBath	SaleCondition
YearRemodel	BedroomAbvGr	SalePrice

2.1 Training and validation samples.

From the eligible samples, 70% of the data is randomly samples to be used as the dataset on which model is developed from. This dataset would be refered to as training dataset. The remaining 30% is used as the validation set to evaluate the model performance of predicting sale price on data that is outside the training set.

APPENDIX

A.1 Data quality check

Tables below shows the summary statisics of the numeric variables and it is noted that statistics are within reasonable bounds and appear to be in the units of measure as described in the data dictionary with only 2 rows missing. Also shown are the number of levels or categories in the nominal variables and the number of missing data (0 missing). The data is deemed usable.

Table 3: Data sanity check for numeric variables

	min	Q1	median	Q3	max	mean	sd	n	missing
SID	1	687.50	1472.0	2176.25	2930	1452.1068627	841.7744368	2040	0
LotArea	1700	8400.00	10015.5	12221.00	215245	11100.0955882	7965.2123783	2040	0
OverallCond	1	5.00	5.0	6.00	9	5.5333333	1.0099392	2040	0
YearRemodel	1950	1968.75	1994.0	2004.00	2010	1986.4750000	19.5329819	2040	0
TotalBsmtSF	0	864.00	1064.0	1389.25	6110	1128.7470588	446.6833470	2040	0
GrLivArea	808	1187.00	1495.0	1797.75	5642	1556.7264706	503.2715799	2040	0
BsmtFullBath	0	0.00	0.0	1.00	2	0.4713095	0.5214118	2039	1
BsmtHalfBath	0	0.00	0.0	0.00	2	0.0632663	0.2474984	2039	1
FullBath	0	1.00	2.0	2.00	4	1.6264706	0.5475751	2040	0
HalfBath	0	0.00	0.0	1.00	2	0.4083333	0.5092852	2040	0
BedroomAbvGr	0	3.00	3.0	3.00	6	2.9225490	0.7689929	2040	0
TotRmsAbvGrd	3	6.00	6.0	7.00	15	6.6112745	1.4991809	2040	0
GarageArea	0	390.00	487.5	602.25	1488	504.9240196	203.5675913	2040	0
MoSold	1	4.00	6.0	8.00	12	6.2235294	2.6746105	2040	0
YrSold	2006	2007.00	2008.0	2009.00	2010	2007.7857843	1.3149064	2040	0
SalePrice	58500	141000.00	176000.0	228000.00	755000	196257.2794118	80242.3169041	2040	0

Table 4: Data sanity check for nominal variables

	# Unique	n	missing
LotConfig	5	2040	0
Neighborhood	22	2040	0
BldgType	5	2040	0
HouseStyle	8	2040	0
KitchenQual	5	2040	0
SaleCondition	2	2040	0

A.2 R code

```r
knitr::opts_chunk$set(echo = TRUE, tidy.opts = list(width.cutoff = 60),
    tidy = TRUE)
ames <- readr::read_delim(file = "ames_housing_data.csv", delim = ",")
# chamge from scientic notations, to restore to default
# options(scipen = 0)
options(scipen = 999)
library(magrittr)
LivingAreaCutoff <- 800
# Adding drop conditions varaible insert dummy variable to
# code SaleCondition being either equal to 'Normal' or
# 'Partial'
ames$Sale_NrmPar <- ifelse(ames$SaleCondition == "Normal" | ames$SaleCondition ==
    "Partial", 1, 0)
ames$DropCondition <- ifelse(ames$Zoning != "RL", "01: Not LowDensityZone",
    ifelse(ames$Sale_NrmPar == 0, "02: Not Normal/Partial Sale",
        ifelse(ames$Street != "Pave", "03: Street Not Paved",
            ifelse(ames$GrLivArea < LivingAreaCutoff, "04: Less than 800 SqFt",
                "99: Eligible Sample"))))

# Waterfall
waterfall <- ames %>% dplyr::group_by(DropCondition) %>% dplyr::summarise(counts = n())

# Print waterfall table
knitr::kable(waterfall, align = c("l", "r"), caption = "Drop waterfall")
# Define training portion of the data
trainPercent <- round(0.7, 1)
# Columns if interest
colsofinterest <- c("SID", "LotArea", "LotConfig", "Neighborhood",
    "BldgType", "HouseStyle", "OverallCond", "YearRemodel", "TotalBsmtSF",
    "GrLivArea", "BsmtFullBath", "BsmtHalfBath", "FullBath",
    "HalfBath", "BedroomAbvGr", "KitchenQual", "TotRmsAbvGrd",
    "GarageArea", "MoSold", "YrSold", "SaleCondition", "SalePrice")

colsmatrix <- matrix(colsofinterest[2:length(colsofinterest)],
    ncol = 3)
knitr::kable(colsmatrix, caption = "Variables of interest")
SampleFrame <- ames %>% dplyr::filter(DropCondition == "99: Eligible Sample") %>%
    dplyr::select_(.dots = colsofinterest)

train <- dplyr::sample_n(SampleFrame, size = trainPercent * nrow(SampleFrame),
    replace = F, set.seed(2000))
train <- train %>% dplyr::arrange(SID)
Validation <- dplyr::sample_n(SampleFrame, size = (1 - trainPercent) *
    nrow(SampleFrame), replace = F, set.seed(2000))
Validation <- Validation %>% dplyr::arrange(SID)
library(mosaic)
sanitycheck <- do.call(rbind, dfapply(SampleFrame, favstats,
    select = is.numeric))
knitr::kable(sanitycheck, caption = "Data sanity check for numeric variables")
sanitycheckcharacter <- select(SampleFrame, colnames(SampleFrame[1,
    sapply(SampleFrame, class) == "character"]))
```

```r
library(purrr)
UniqueVals <- sanitycheckcharacter %>% map(unique)
# s <-
# data.frame(names(tst),sapply(tst,function(x){paste(x,collapse
# = ',')}),row.names = NULL)
Counts <- data.frame(sapply(UniqueVals, length), do.call(rbind,
    dfapply(sanitycheckcharacter, length, select = is.character)),
    do.call(rbind, dfapply(sanitycheckcharacter, n_missing, select = is.character)),
    row.names = names(UniqueVals))
colnames(Counts) <- c("# Unique", "n", "missing")

knitr::kable(Counts, caption = "Data sanity check for nominal variables",
    align = c("l", "r", "r", "r"))
```