ASSIGNMENT 2
GROUP COURSEWORK
ON HOUSE_DATA

Submitted in
partial fulfillment
of the module

MA 321
APPLIED STATISTICS

Submitted By
GROUP F

Daniel Salnikov 1900898

Dilpa Rao 1906319

Vasileios Panagaris 1901009

Viraj Kumar Dewangan 1901181

GUIDED BY -
Dr. Fanlin Meng
Dr. Stella Hadjiantoni

Date of Submission -
27 March 2020

Contents

Appendix

Abstract

This report studies if location and other features like area, garage, neighborhood, number of bedrooms, etc. have a statistical relationship with the price, size, novelty and other variables of a house. It does this by analyzing US census data from 2006 to 2010. It fits logistic regression, linear discriminant analysis, gradient tree boosting, random forests and other linear models to study the relationship between price, location and features. After the analysis, the report provides significant evidence that location matters a LOT in the Real State Market.

Group Work Contribution

The assignment taskwork was carried out in group with **Daniel Salinkov** as the **group leader** who guided and supervised the entire project.

Word Count: 3393 words

Page Count without Cover Page & without Appendix: 10 pages

1 Introduction

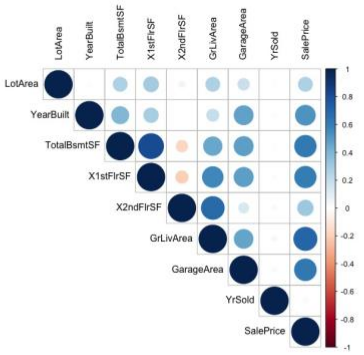
This project will analyze if a house's price, size and age are able to provide information on the house's location and condition. This study will fit 2 classification models to classify the house's **OverCond**. These models are Logistic multinomial regression and Linear Discriminant Analysis (LDA). Later it will explore how features (age, size, quality) estimate **SalePrice**. The report analyzes regressing **SalePrice** on **GrLivArea** among other features by gradient tree boosting and random forests techniques. Finally, the report classifies the house's location as a function of price by use of a Support Vector Machine (SVM) and LDA. These analyses are employed to account for differences on the mean of **SalePrice** and what may be a possible model to estimate **SalePrice** and its distribution. All to answer how much does location matter in real state.

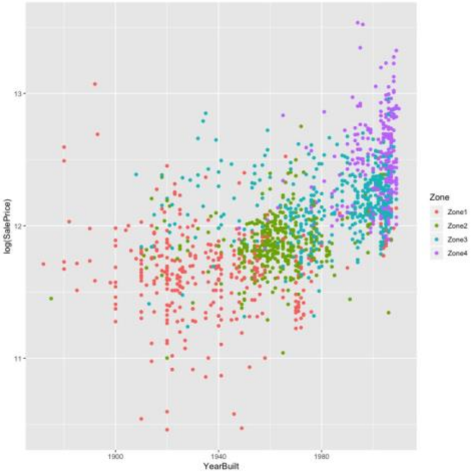
2 Preliminary Analysis

The data consist of 1460 observations of 51 variables. There are 12 numerical variables and 39 categorical variables. The following varibles are missing most of their values and/or have no variability among them. Alley, Fence, PoolQC, PoolArea, MiscFeature, MiscVal, Utilities, OverallCond, LowQualFinSF ,LotFrontage. OverallCond was transformed into 3 categories:

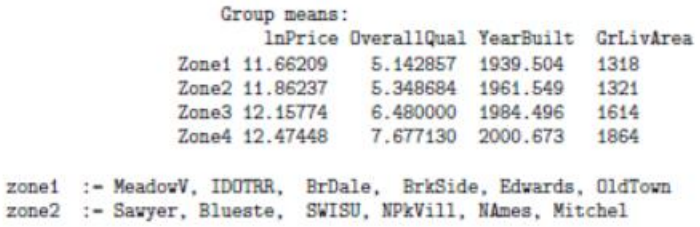
$$Average = 1130; Good = 299; P\ oor = 31$$

The numerical data has the following correlation plot matrix:





This project studies **SalePrice** distribution within the city. The numerical predictors used for adjusting models were **SalePrice**, **GrLivArea**, **YearBuilt**, **OverQual**. Also, some categories that affect the **SalePrice** are **Zone**, **OverCond**, **BldgType**. The variable Zone divides the 25 Neighborhoods into 4 Zones regarding the average value of **SalePrice.** This will study if and how Location affects **SalePrice.**



```
          Group means:
             lnPrice OverallQual YearBuilt  GrLivArea
      Zone1 11.66209    5.142857  1939.504  1318
      Zone2 11.86237    5.348684  1961.549  1321
      Zone3 12.15774    6.480000  1984.496  1614
      Zone4 12.47448    7.677130  2000.673  1864

zone1  :- MeadowV, IDOTRR,  BrDale,  BrkSide, Edwards, OldTown
zone2  :- Sawyer, Blueste,  SWISU, NPkVill, NAmes, Mitchel
```

zone3: = Crawfor, Gilbert, ClearCr, CollgCr, Blmngtn, NWAmes, SawyerW
zone4: = NoRidge, NridgHt, StoneBr, Timber, Veenker, Somerst

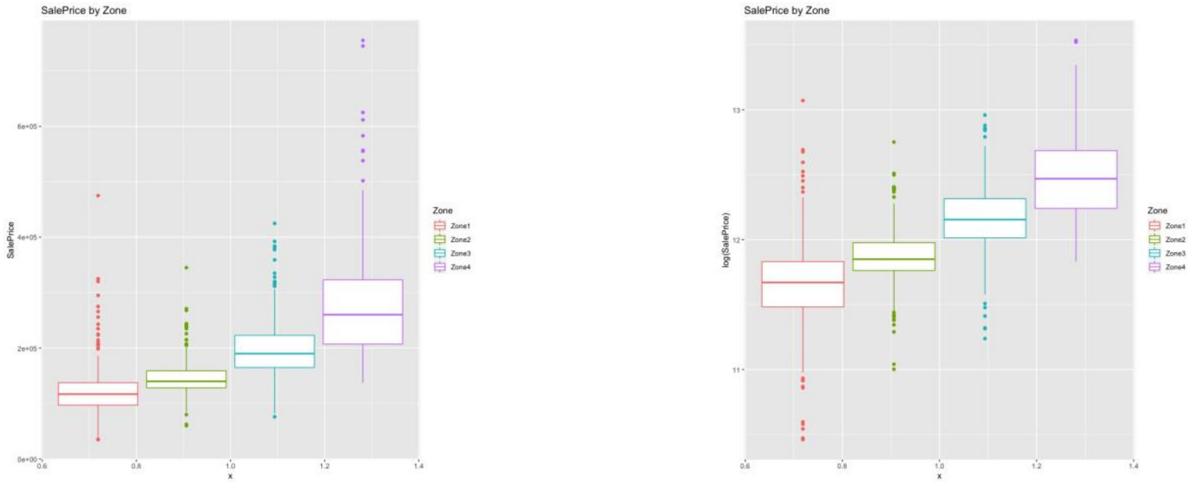


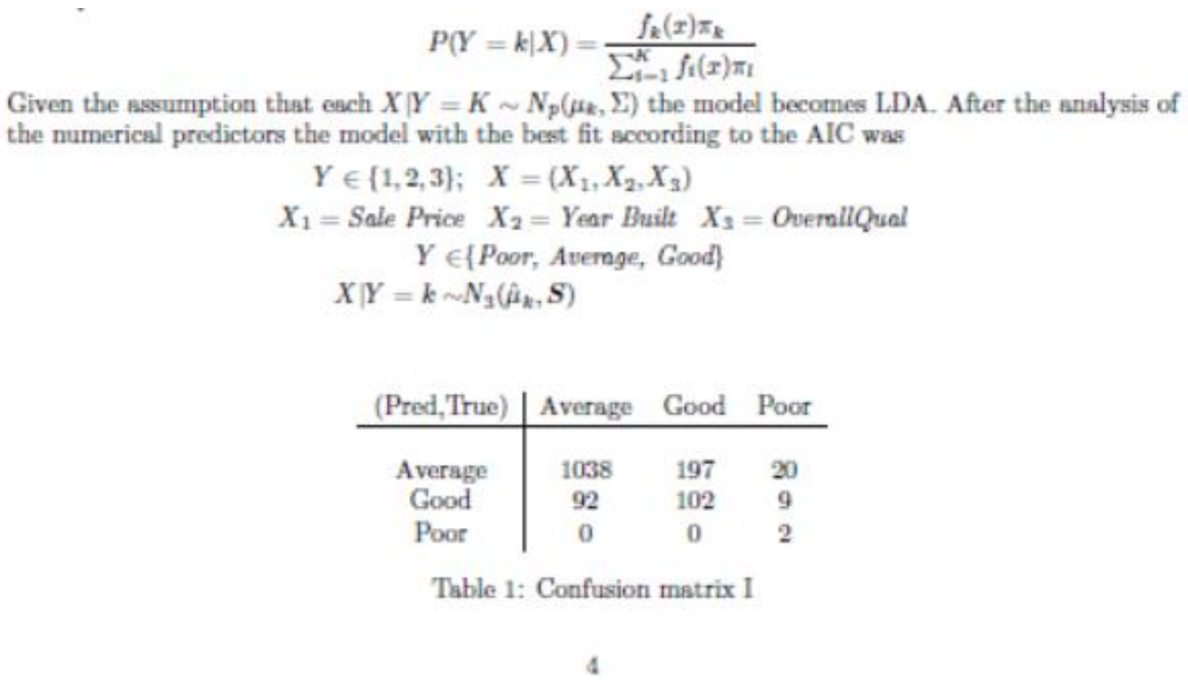
Figure 1: Price distribution between zones.

The figure illustrates the different **SalePrice** distributions regarding **Zone**. The numerical data is skewed and has a lot of outliers; thus, it is sensible to log transform these data to obtain more symmetric distributions.

3 Analysis

3.1 Linear Discriminant Analysis

The first question the project aimed to answer was how to fit the logistic regression and Linear Discriminant Analysis (LDA) and look at the fit through the confusion matrix. To adjust the LDA it was necessary to only use numerical features because this model assumes that the predictors are normally distributed; thus, using categorical data as a predictor would be inappropriate as it violates the key assumptions of the model.



$$P(Y=k|X)=\frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

Given the assumption that each $X|Y=K \sim N_p(\mu_k,\Sigma)$ the model becomes LDA. After the analysis of the numerical predictors the model with the best fit according to the AIC was

$$Y \in \{1,2,3\}; \quad X=(X_1,X_2,X_3)$$
$$X_1 = Sale\ Price \quad X_2 = Year\ Built \quad X_3 = OverallQual$$
$$Y \in \{Poor,\ Average,\ Good\}$$
$$X|Y=k \sim N_3(\hat{\mu}_k, \boldsymbol{S})$$

(Pred,True)	Average	Good	Poor
Average	1038	197	20
Good	92	102	9
Poor	0	0	2

Table 1: Confusion matrix I

3.2 Logistic Regression

The logistic regression model was fitted using a multinomial likelihood for the 3 different classes for the condition variable. Given that most of the observations had a house condition of average this class was chosen as the base; thus, the model estimated the log-odds of not being Average.

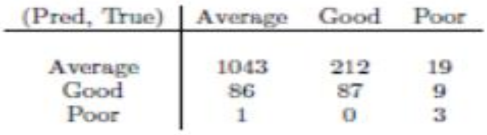
Model II:

Y ~ Mult3(n, p1, p2)

The fitted model used the same 3 numerical predictors as the LDA, but added the categorical variable Building Type to see if it affected the building's condition. There are 5 distinct building types. The fitted model had the following summary:

$$\ln\left(\frac{P(Y=Good)}{P(Y=Average)}\right) = 76.154 + 4.48e - 06 SalePrice - 0.1231\, OverallQual - 0.039\, YearBuilt + BldgType$$

$$\ln\left(\frac{P(Y=Poor)}{P(Y=Average)}\right) = -0.95 - 2.723 SalePrice + 0.384\, OverallQual - 1.539\, YearBuilt + BldgType$$



(Pred, True)	Average	Good	Poor
Average	1043	212	19
Good	86	87	9
Poor	1	0	3

Table 2: Confusion Matrix II

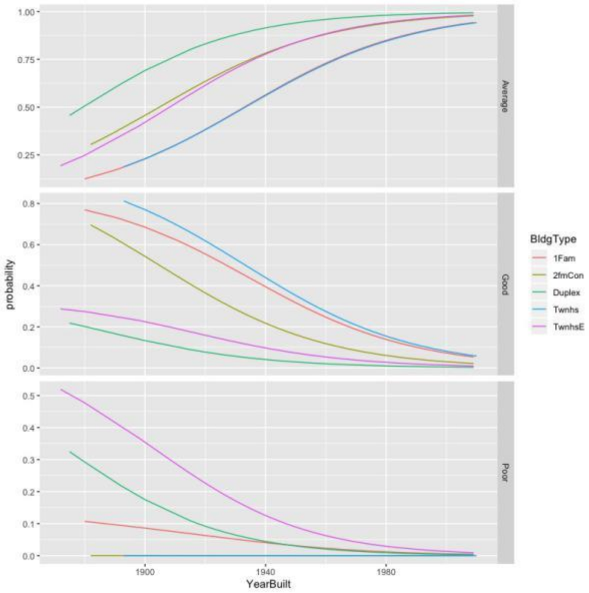


Figure 2: Predicted probabilities.

The plot illustrates how **OverCond** varies differently for each **BldgType**. The older the house the more likely it is in a poor condition.

3.3 Gradient Boosting

Predicting house prices is a regression problem because the sale Price could be any positive number; nevertheless, the geographic location could give a hint of where are prices typically higher than in other places. Another problem is that the Sale prices have heavy tails and a bunch of outliers; thus, it is sensible to take the logarithm of the sale price before fitting the model. After transforming the prices, it is possible to think of the standard errors as normally distributed; therefore, one is able to regress the **log (SalePrice)** on **GrLivArea, YearBuilt** and **OverQual** to estimate **log (SalePrice)**. **Gradient tree boosting** with squared error loss was fitted to these data by boosting T iterations.

GBM for SalePrice:

$$Y = \ln(SalePrice); \quad X = (X_1, X_2, X_3)$$

$$X_1 = \ln(GrLivArea) \; X_2 = \ln(YearBuilt) \; X_3 = OverQual$$

$$l(y, f(x)) = \frac{1}{2}(y - f(x))^2; \quad r_i = y_i - f(x_i)$$

$$f_t(x) = f_{t-1}(x) + \sum_{j=1}^{J} \lambda_j I(x \in R_j)$$

$$\hat{y} = f_T(x); \quad \widehat{SalePrice} = \exp(\hat{y})$$

$\hat{f}$ is the boosted sum of trees that results from applying the gradient boos algorithm with T iterations. The R library for implementing this analysis is gbm. Note that squared error loss makes the model adjust a least squares fit to the residuals at each iteration. Even though this is not

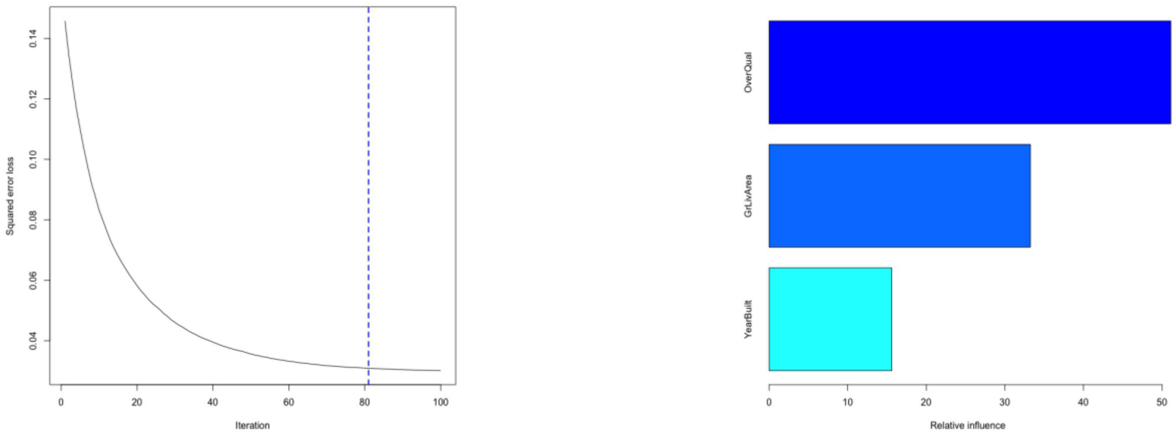


Figure 3: Squared error loss of gradient tree boosting.

The squared error loss did decrease by applying the boosting algorithm with a shrinkage value of 0.1. This model had R^2 = 0:811 and M SE = 0:03 for T = 80 iterations.

3.4 Random Forest

The method of random forest allows to see how the numerical predictors may be used to identify the predictors with a stronger relationship to the response. Given that this is a regression tree with a heavily skewed response **SalePrice** with a fat tail the model was fitted to the response's logarithm.

$$Y = \ln(\mathit{SalePrice}); \quad X = (X_1, X_2, X_3)$$
$$X_1 = \mathit{GrLivArea} \ \ X_2 = \mathit{YearBuilt} \ \ X_3 = \mathit{OverQual}$$
$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x)$$
$$\hat{y} = \hat{f}(x_1, x_2, x_3)$$
$$\widehat{\mathit{SalePrice}} = \exp(\hat{y})$$

The randomForest library uses $B = 500$ as the default number of random trees to be fitted to the data. This choice did not affect performance, so the project did not tweak this value. A random forest fitted to the whole numerical data gave the following result:

```
    Call:
 randomForest(formula = log(SalePrice) ~ ., data = data_num)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 0.02411691
                    % Var explained: 84.88
```

A random forest fitted to the 3 predictors with the highest correlation to the **SalePrice** resulted in:

```
    Call:
 randomForest(formula = log(SalePrice) ~ YearBuilt +
 OverQual + GrLivArea, data = data_num)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 1

          Mean of squared residuals: 0.03046082
                    % Var explained: 80.9
```

Thus, the additional predictors improve the fitted model marginally.

3.5 Support Vector Machine

The support vector machine can classify data that have overlap among the classes; thus, it was useful to classify the Zone of the city. There clearly is a relationship between the price, zone and age of the property. This information is useful to group the house by zone rather than the whole city together or by neighborhood. The R package used was e1071 and the kernel function was:

$$k(x, \tilde{x}) = \exp(\sigma\|x - \tilde{x}\|^2)$$

$Y \in \{1, 2, 3, 4\}$ zone of the city classified by $X_1 = \ln(SalePrice)$ and $X_2 = YearBuilt$ The fitted model resulted in the confusion matrix below; also, the plot illustrates the complexity of classifying data under

(Pred, True)	Zone1	Zone2	Zone3	Zone4
Zone1	188	36	19	0
Zone2	57	238	71	4
Zone3	20	29	255	96
Zone4	1	1	30	123

Table 3: Confusion Matrix

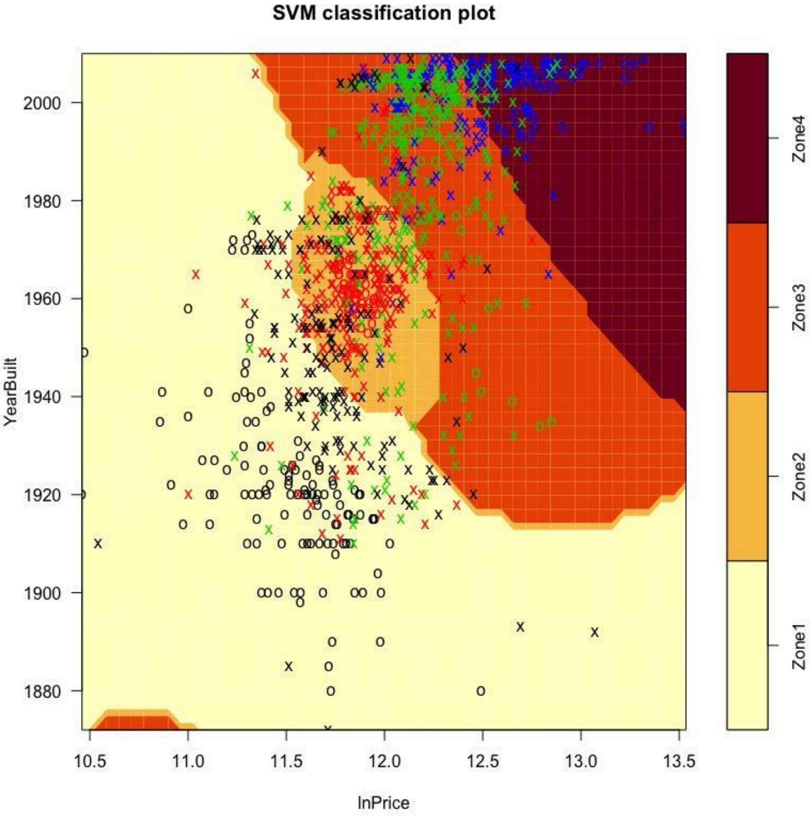


Figure 4: Zone classification.

Even with overlap among classes (Zone) there still is a trend that the **SalePrice** is associated with a specific Zone. This model classifies correctly 71:23% of the data.

3.6 Location, Location, Location

Given that the best classifier would be a Bayes decision rule it was interesting to adjust an LDA to the zone of the house to see if the **SalePrice** could determine an efficient way of classifying the Zone of a property. Nevertheless, to properly predict house prices one must consider the fat tail. A fair assumption is to think of the **SalePrice** as being log-normal distributed.

$$SalePrice \sim logNormal(\mu, \sigma^2) \tag{1}$$

To see of there is evidence that one may classify the Zone by the **SalePrice**.

$$Y \in \{1,2,3,4\}; \quad X = (X_1, X_2, X_3)$$

$$X_1 = \ln(Sale\ Price) \quad X_2 = Year\ Built \quad X_3 = OverallQual$$

$$Y \in \{Zone1,\ Zone2,\ Zone3,\ Zone4\}$$

$$X|Y = k \sim N_3(\hat{\mu}_k, \boldsymbol{S})$$

The fitted LDA provides evidence of difference in the means of the numerical quantities.

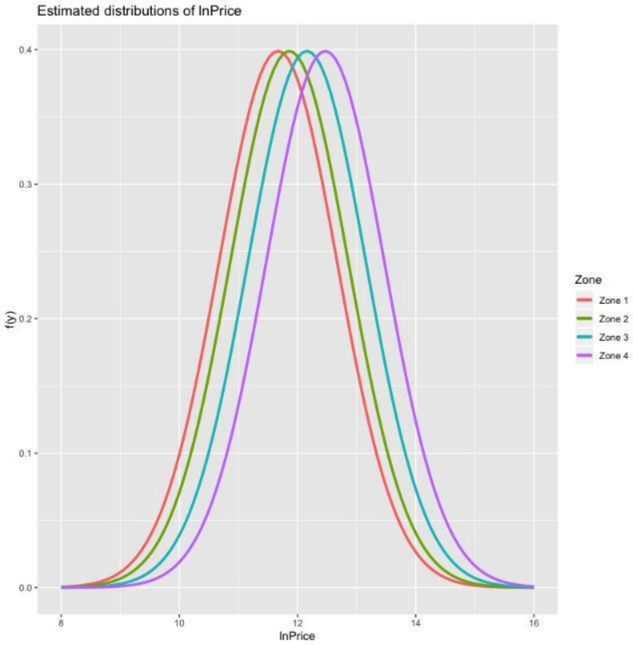


Figure 5: Estimated distributions.

By assuming (1) it is possible to estimate the distribution of $\epsilon = \ln(SalePrice) - \ln(\widehat{SalePrice})$ as normal; thus, a linear model may be fitted to these residuals; furthermore, it is possible to enhance the linear models by employing the gradient boosting algorithm on the residuals because square error loss is assumed. It is possible to estimate the distribution and value of property by accounting for difference in location Zone and the numerical predictors **GrLivArea, YearBuilt, OverallQual**. One possibility is maximum likelihood inference; thus, the approximate normal distributions are parametrized by the mean and variance maximum likelihood estimators for each Zone. The distribution of ln (SalePrice) may be approximated by a mixture of normal distributions, given these assumptions.

$$Y = \ln(SalePrice); \quad X = (X_1, X_2, X_3); \quad p_k = P(Zone = k) \quad k = 1, \cdots, 4$$

$$X_1 = \ln(GrLivArea) \quad X_2 = \ln(YearBuilt) \quad X_3 = OverQual$$

$$f(y) = \sum_{k=1}^{4} p_k f_k(y)$$

$$Y|Z = k \sim N(\mu_k, \sigma_k^2)$$

$$\mu_k = X_k \beta_k; \quad \sigma_k^2 > 0$$

This model was tested by validation set approach with 80% of the data set used for training and 20% for testing. The confusion matrix for the test set is:

(Pred, True)	Zone1	Zone2	Zone3	Zone4
Zone1	48	8	3	0
Zone2	20	53	15	1
Zone3	5	19	57	15
Zone4	2	0	7	39

Table 4: Confusion matrix.

The estimated probabilities are used with the predicted mean to compute a estimation for **SalePrice**. Each $\hat{y}_k$ is estimated by use of the specific **Zone** model.

$$\hat{y} = \hat{p}_1\hat{y}_1 + \hat{p}_2\hat{y}_2 + \hat{p}_3\hat{y}_3 + \hat{p}_4\hat{y}_4$$

The model using linear estimates (linear regression) had a $MSE = 0.45$ and $R^2 = 0.73$; furthermore, when using gradient boosting with square loss the model computed $MSE = 0.04$ and $R^2 = 0.76$. This model takes into account uncertainty regarding the **Zone**. The model is useful to study a propertie's price and how it compares to other properties in its distribution. This allows to compute more rigorous prices based on observed data.

4 Discussion

The fitted classification models were tested by LOOCV and validation set approach. The validation set was sampled from 20% of the data. The first comparison compares Linear discriminant analysis and multinomial logistic regression fitted the **OverCond.**

Model	$\widehat{err}_{\text{test}}$	$\widehat{err}_{\text{LOOCV}}$
LDA	0.2054795	0.2226027
logit	0.1986301	0.2184932

Table 5: Model comparison.

LDA I classifies Zone without use of **SalePrice** and LDA II incorporates **SalePrice** to the model.
There is evidence that **SalePrice** does classify the Zone, (i.e. location matters).

Model Model	$\widehat{err}_{\text{test}}$	$\widehat{err}_{\text{LOOCV}}$
LDA I	0.3253425	0.3294521
LDA II	0.2876712	0.3417808

Table 6: Zone Classification.

Zone classification is sensible because there are outliers in each neighborhood, this could be over or under priced properties; nevertheless, for data that are skewed and asymmetric the LDA was able to classify a house's Zone and **OverCond** based on **SalePrice.** It should be noted that these data are heavily skewed; thus, assumptions about symmetry, normality, and especially skinny should be handled with a grain of salt because much of the numerical data exhibits a heavy tail; also, most of the categorical data is either missing or difficult to interpret. Because of that most of the models fitted used numerical predictors adjusted to more symmetrical distribution. This violates less the assumptions of the models. Nevertheless, the approximations should be taken with a grain of salt. The boosted model statistics MSE = 0.03, R^2 evidence the relationship between **SalePrice** and **GrLivArea, YearBuilt, Zone, OverQual**.

5 Conclusion

There is a saying in real state "Location, location, location". This report provides some evidence that indeed **Zone** influences the **SalePrice** (F = 532.4, df=3, p ¡ 2.2e-16) even more than **OverCond** (F = 46.77, df=2, p ¡ 2.2e-16); nevertheless, both of these categories have an effect on **SalePrice**, thus **SalePrice** is useful to classify a house's condition and location. This analysis may help people comparing house prices. It is important to account for the skewed distribution of prices and how this affects the precision of estimates; nevertheless, the real state saying "Location, location, location" is statistically significant.

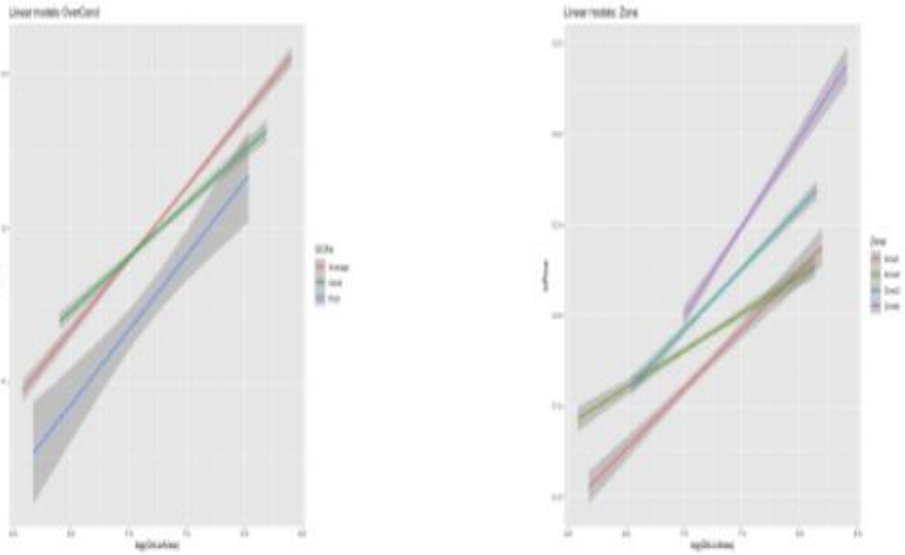


Figure 6: Effect of Zone and Condition.

Notice that the **Zone** models's slope vary more.

References

[1] "Multinomial Logistic Regression." IDRE Stats, 2014, stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/.

[2] Ridgeway, Greg. Gbm Package, r-project-library, 14 Jan. 2019, 127.0.0.1:10423/library/gbm/-doc/gbm.pdf.

[3] Science, ODSC - Open Data. "Build a Multi-Class Support Vector Machine in R." Medium, Medium, 15 Nov. 2018, medium.com/@ODSC/build-a-multi-class-support-vector-machine-in-r-abcdd4b7dab6.

[4] Analytics, Perceptive. "How to Implement Random Forests in R." R, 9 Jan. 2018, www.r-bloggers.com/how-to-implement-random-forests-in-r/.

Appendix

Algorithm 1 Gradient tree boosting algorithm

```
Loss function:
    $$l(y, f(x)) = \frac{1}{2}(y - f(x))^2$$
    $$r = -\frac{\partial l}{\partial f(x)}$$
Initialize
    $$f_0(x) = \arg_\lambda \min \sum_{i=1}^{n} l(y_i, \lambda)$$
Boost regression trees:
for $t = 1, \cdots, T$ do
    i) for $i = 1, \cdots, n$ do
        $$r_{it} = y_i - f_{t-1}(x_i)$$
    end
    ii) Fit regression tree to the residuals/gradient
    Each tree has $J$ terminal nodes, this could change every iteration but for simplicity this project kept them equal.
        $$R_{jt} \in \{R_{1t}, \cdots, R_{Jt}\}$$
        $$\hat{g}(x) = E(r|x) = \hat{r} \quad \textit{Region mean.}$$
    iii) Find new optimal nodes.
    for $j = 1, \cdots, J$ do
        $$\lambda_j = \arg_\lambda \min \sum_{x_i \in R_j} \frac{1}{2}(r_{it} + \lambda)^2$$
    end
    iv) Update:
        $$f_t(x) = f_{t-1}(x) + \sum_{j=1}^{J} \lambda_j I(x \in R_j)$$
end
Result: Weighted sum of trees:
    $$\hat{f}(x) = f_T(x)$$
```

R code:

```
    ##Load the data
dirtyData <- read.csv("house_data.csv")
dirtyData <- dirtyData %>% filter(!is.na(OverallCond)) %>% filter(!is.na(SalePrice))
meanSaled <- dirtyData %>% group_by(Neighborhood) %>%
  summarise(meanS = mean(SalePrice)) %>% arrange(meanS)
```

```
##Some variables are purely missing values it is wise to
get rid of these because they do not provide information
drops <- c("Alley", "Fence",
"PoolQC", "PoolArea", "MiscFeature", "Id", "MiscVal",
"Utilities", "OverallCond", "LowQualFinSF", "LotFrontage")
##The majority of the predictors are categorical; thus, it is a
good strategy to get the set of quantitative predcitors
##and analyse each predicotr to idealize its rlationship with the
other variables and how it might inform a model for a
##specific response variable
preds_num <- c( "LotArea", "OverQual", "YearBuilt", "MasVnArea", "TotalBsmtSF",
"X1stFlrSF", "X2ndFlrSF",
"GrLivArea", "GarageArea", "YrSold", "SalePrice")
data_num <- dirtyData[, names(dirtyData) %in% preds_num] %>% filter(!is.na(SalePrice))
data_num$OverQual <- wor_data$OverallQual

OverCondF <- c(1:length(dirtyData$OverCond))
for(i in 1:length(dirtyData$OverallCond)){
  if(dirtyData$OverallCond[i] <= 3){
    OverCondF[i] = "Poor"
  } else {
    if(dirtyData$OverallCond[i] <= 6){
      OverCondF[i] = "Average"
    } else {
      OverCondF[i] = "Good"
    }
  }
}
dirtyData$OCFa <- as.factor(OverCondF)
##Explore correlation between
quantitative variables and overlaps among categorical data.
ggplot(wor_data, aes(x = 1, y = log(SalePrice), color = Zone))+
  geom_boxplot()+
  ggtitle("SalePrice by Zone")
hist(wor_data$lnPrice, freq = FALSE)
hist(wor_data$SalePrice, freq = FALSE)

corrplot(cor(data_num), tl.col="black", type = "upper")

##Question II
##Logistic regression for the condition of the house

dirtyData <- dirtyData %>% filter(!is.na(SalePrice))
##Fit the model using glm library
dirtyData$OCFac <-relevel(dirtyData$OCFa, ref = "Average")
logit1 <- multinom(OCFac ~ OverallQual + YearBuilt + BldgType, data = dirtyData)
summary(logit1)
BIC(logit1)
logit1_pred <- fitted(logit1)
predLog1 <- predict(logit1, dirtyData, type = "class")
table(predLog1, dirtyData$OCFac)


 ##Fit using LDA check confussion matrix
 lda1 <- lda(OCFa ~ log(SalePrice) + YearBuilt + OverallQual + log(GrLivArea), data = dirtyData)
 lda1_predict <- predict(lda1)$class
 table(lda1_predict, dirtyData$OCFa)

 ##LOOCV
```

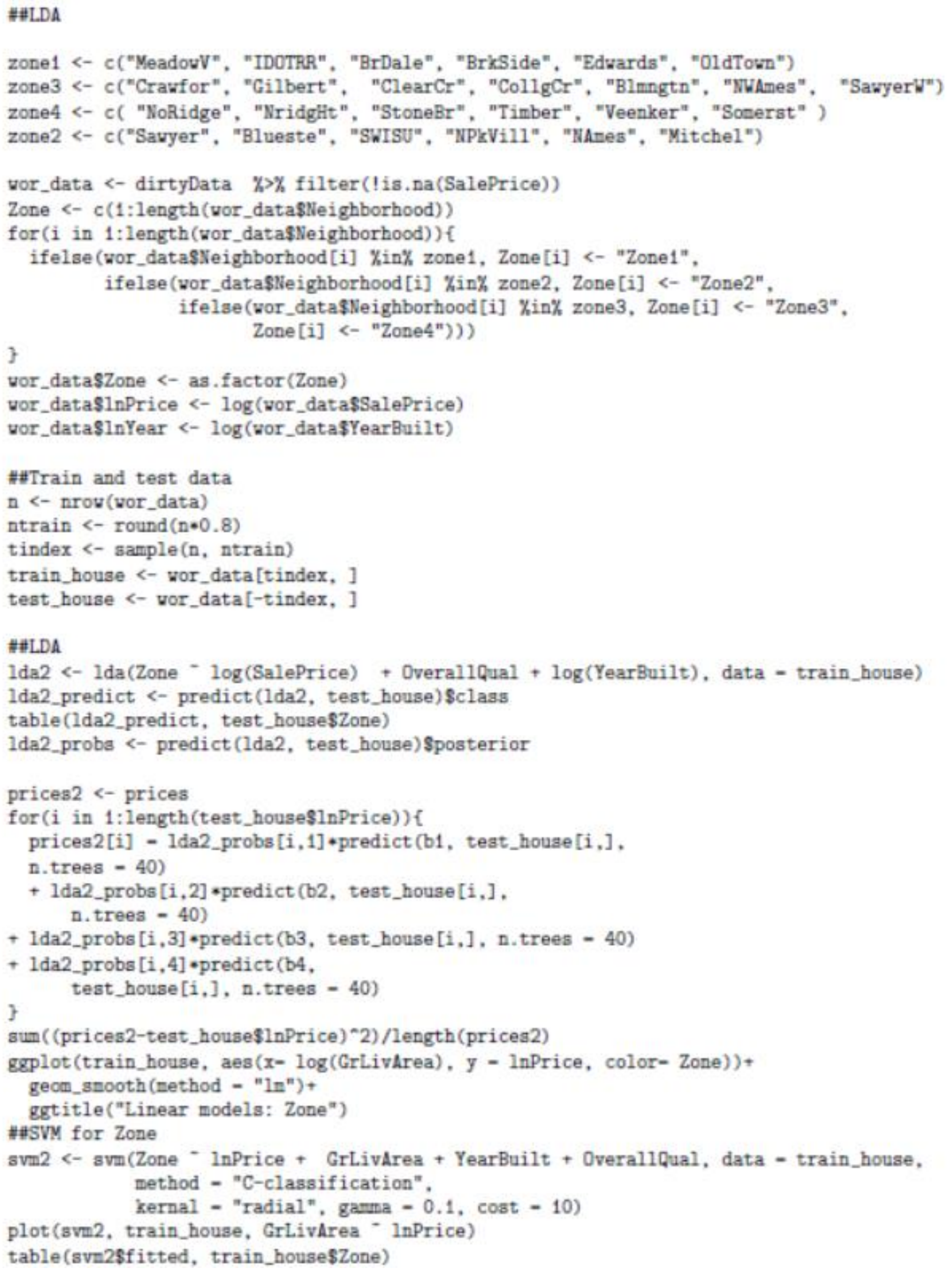
```
err <- rep(0, 1460)
err2 <- rep(0, 1460)
for(i in 1:1460){
  out <- i
  test <- dirtyData[out, ]
  train <- dirtyData[-out, ]
  logist <- multinom(OCFac ~ SalePrice + OverallQual + YearBuilt + BldgType, data = train)
  pred <- predict(logist, test, type = "class")
  ifelse(pred == test$OCFac, err[i] <- 0, err[i] <- 1)
}
for(j in 1:1460){
  out <- j
  test <- dirtyData[out, ]
  train <- dirtyData[-out, ]
  ldas <- lda(OCFa ~ SalePrice + OverallQual + YearBuilt, data = train)
  pred <- predict(ldas, test, type = "class")
  ifelse(pred$class == test$OCFa, err2[j] <- 0, err2[j] <- 1)
}

##Predicting house prices and model assessment

##The library in use is GBM written by Greg Ridgeway
##gradient boosting regression trees with square loss
boost1 <- gbm(log(SalePrice) ~ .,
distribution = "gaussian", data = data_num)
gbm.perf(boost1, plot.it = TRUE)
summary(boost1)


##Random Forest
forest1 <- randomForest(log(SalePrice) ~ YearBuilt + OverQual + GrLivArea, data = data_num)
forest1
##Missclasification error estimate
nh <- nrow(dirtyData)
nhtrain <- round(nh*0.8)
hindex <- sample(nh, nhtrain)
train_OCFac <- dirtyData[hindex, ]
test_OCFac <- dirtyData[-hindex, ]
##LDA missclaf error
lda3 <- lda(OCFa ~ log(SalePrice) + YearBuilt + OverallQual, data = train_OCFac)
lda3_predict <- predict(lda3, newdata = test_OCFac)$class
table(lda3_predict, test_OCFac$OCFa)
##Logistic error
train_OCFac$OCFac <- relevel(train_OCFac$OCFa, ref = "Average")
test_OCFac$OCFac <- relevel(test_OCFac$OCFa, ref = "Average")
logit2 <- multinom(OCFac ~ SalePrice + OverallQual + YearBuilt + BldgType, data = train_OCFac)
summary(logit2)
BIC(logit2)
logit2_pred <- fitted(logit2)
predLog2 <- predict(logit2, test_OCFac, type = "class")
table(predLog2, test_OCFac$OCFac)
```

```
##Classify location based on price, size and other key aspects.
```



```
##LDA

zone1 <- c("MeadowV", "IDOTRR", "BrDale", "BrkSide", "Edwards", "OldTown")
zone3 <- c("Crawfor", "Gilbert",  "ClearCr", "CollgCr", "Blmngtn", "NWAmes",  "SawyerW")
zone4 <- c( "NoRidge", "NridgHt", "StoneBr", "Timber", "Veenker", "Somerst" )
zone2 <- c("Sawyer", "Blueste", "SWISU", "NPkVill", "NAmes", "Mitchel")

wor_data <- dirtyData  %>% filter(!is.na(SalePrice))
Zone <- c(1:length(wor_data$Neighborhood))
for(i in 1:length(wor_data$Neighborhood)){
  ifelse(wor_data$Neighborhood[i] %in% zone1, Zone[i] <- "Zone1",
         ifelse(wor_data$Neighborhood[i] %in% zone2, Zone[i] <- "Zone2",
                ifelse(wor_data$Neighborhood[i] %in% zone3, Zone[i] <- "Zone3",
                       Zone[i] <- "Zone4")))
}
wor_data$Zone <- as.factor(Zone)
wor_data$lnPrice <- log(wor_data$SalePrice)
wor_data$lnYear <- log(wor_data$YearBuilt)

##Train and test data
n <- nrow(wor_data)
ntrain <- round(n*0.8)
tindex <- sample(n, ntrain)
train_house <- wor_data[tindex, ]
test_house <- wor_data[-tindex, ]

##LDA
lda2 <- lda(Zone ~ log(SalePrice)  + OverallQual + log(YearBuilt), data = train_house)
lda2_predict <- predict(lda2, test_house)$class
table(lda2_predict, test_house$Zone)
lda2_probs <- predict(lda2, test_house)$posterior

prices2 <- prices
for(i in 1:length(test_house$lnPrice)){
  prices2[i] = lda2_probs[i,1]*predict(b1, test_house[i,],
  n.trees = 40)
  + lda2_probs[i,2]*predict(b2, test_house[i,],
      n.trees = 40)
+ lda2_probs[i,3]*predict(b3, test_house[i,], n.trees = 40)
+ lda2_probs[i,4]*predict(b4,
      test_house[i,], n.trees = 40)
}
sum((prices2-test_house$lnPrice)^2)/length(prices2)
ggplot(train_house, aes(x= log(GrLivArea), y = lnPrice, color= Zone))+
  geom_smooth(method = "lm")+
  ggtitle("Linear models: Zone")
##SVM for Zone
svm2 <- svm(Zone ~ lnPrice +  GrLivArea + YearBuilt + OverallQual, data = train_house,
            method = "C-classification",
            kernal = "radial", gamma = 0.1, cost = 10)
plot(svm2, train_house, GrLivArea ~ lnPrice)
table(svm2$fitted, train_house$Zone)
```