```r
OSR2 <- function(predictions, train, test) {
  SSE <- sum((test - predictions)^2)
  SST <- sum((test - mean(train))^2)
  r2 <- 1 - SSE/SST
  return(r2)
}

#Read datasets
Listing = read.csv("Zip_MedianListingPricePerSqft_AllHomes.csv",
 stringsAsFactors=FALSE)
Rental = read.csv("Zip_MedianRentalPricePerSqft_AllHomes.csv",
 stringsAsFactors=FALSE)
Income = read.csv("Personal_Income_Tax_Statistics_By_Zip_Code.csv",
 stringsAsFactors=FALSE)
SAT = read.csv("sat_score_with_ZIP.csv", stringsAsFactors=FALSE)
Review = read.csv("CA_Review.csv", stringsAsFactors=FALSE)

head(Review)
head(Listing)
head(Rental)
head(Income)
head(SAT)

#Select California ones
Rental <- Rental[Rental$State == "CA", ]
Listing <- Listing[Listing$State == "CA", ]
Income <- Income[Income$State == "CA", ]

#Average last 36 months prices
Listing$AvgListing <- rowMeans(Listing[,(ncol(Listing)-35):ncol(Listing)],
 na.rm = TRUE)
Rental$AvgRental <- rowMeans(Rental[,(ncol(Rental)-35):ncol(Rental)], na.rm =
 TRUE)

#Some more cleaning
Listing$ZipCode <- Listing$RegionName
Rental$ZipCode <- Rental$RegionName
Income$ZipCode <- as.integer(Income$Zip.Code)

Listing <- Listing[,c("ZipCode", "City", "CountyName", "AvgListing")]

Rental <- Rental[,c("ZipCode","AvgRental")]

Income <- Income[Income$Taxable.Year == "2017",c("ZipCode","CA.AGI")]

library(dplyr)
Listing <- inner_join(Listing, Rental, by = "ZipCode")

Listing <- inner_join(Listing, Income, by = "ZipCode")

nrow(Listing)
```

```r
head(Listing)

#add demographics
Demo = read.csv("demobirth.csv", stringsAsFactors=FALSE)

head(Demo)

Data = inner_join(Listing, Demo, by = c("ZipCode"="Geography"))[,-7]

Data = Data[,-c(2:3)]
Data$Population.Per.Square.Mile..Land.Area. =
 as.numeric(Data$Population.Per.Square.Mile..Land.Area.)

#Add SAT Percentage of Passing Benchmark
SAT <- SAT[,c("Zip", "PctBothBenchmark")]
head(SAT)

SAT$Zip = as.numeric(SAT$Zip)
Data = inner_join(Data, SAT, by = c("ZipCode"="Zip"))
head(Data)
nrow(Data)

#Add AirBnB review polarity score
Data = inner_join(Data, Review, by = c("ZipCode"="zip"))
Data = na.omit(Data)
head(Data)
nrow(Data)

# split training into training, testing and validation set
set.seed(122)
train.ids <- sample(nrow(Data), 0.90*nrow(Data))
train <- Data[train.ids,]
test <- Data[-train.ids,]

val.ids <- sample(nrow(train), (10/90)*nrow(train))
val <- train[val.ids,]
train <- train[-val.ids,]

#Linear Regression
lin.mod <- lm(AvgListing ~ . - ZipCode, data = train)
summary(lin.mod)

preds.lm <- predict(lin.mod, newdata = val)

#linear regression validation set MAE, RMSE, OSR2 (un-normalized)
OSR2(preds.lm, train$AvgListing, val$AvgListing)
mean(abs(preds.lm - val$AvgListing)) #MAE
sqrt(mean((preds.lm - val$AvgListing)^2)) #RMSE

library(car)
```

```
vif(lin.mod)

#CART Regression
library(rpart)
library(rpart.plot)
library(caret)

cart.mod <- rpart(AvgListing ~ . - ZipCode,
                  data = train, method = "anova", cp = 0.02, minsplit = 10)

CartPredictions <- predict(cart.mod, newdata=val)

#CART validation set MAE, RMSE, OSR2 (un-normalized)
OSR2(CartPredictions, train$AvgListing, val$AvgListing)
mean(abs(CartPredictions - val$AvgListing))
sqrt(mean((CartPredictions - val$AvgListing)^2))

#cross-validated random forest
set.seed(311)
train.rf.b = train(AvgListing ~ . - ZipCode,
                 data = train,
                 method = "rf",
                 tuneGrid = data.frame(mtry = 1:10),
                 trControl = trainControl(method = "cv", number = 5,
                  verboseIter = TRUE))
train.rf.b
train.rf.b$results

mod.rf.b = train.rf.b$finalModel
predict.rf.b = predict(mod.rf.b, newdata = val)

#random forest performance
OSR2(predict.rf.b, train$AvgListing, val$AvgListing)
mean(abs(predict.rf.b - val$AvgListing))
sqrt(mean((predict.rf.b - val$AvgListing)^2))

#boosting
library(gbm)
mod.boost=gbm(AvgListing ~ . - ZipCode,data = train, distribution = "gaussian",
 n.trees = 1500,
                  shrinkage = 0.1, interaction.depth = 4)
mod.boost

summary(mod.boost)

predict.boost = predict(mod.boost, newdata = val, n.trees = 1500)

#boosting performance
OSR2(predict.boost, train$AvgListing, val$AvgListing)
mean(abs(predict.boost - val$AvgListing))
sqrt(mean((predict.boost - val$AvgListing)^2))
```

```r
#Linear Regression without average rental
lin.mod <- lm(AvgListing ~ . - ZipCode - AvgRental, data = train)
summary(lin.mod)

preds.lm <- predict(lin.mod, newdata = val)

#linear regression validation set MAE, RMSE, OSR2 (un-normalized)
OSR2(preds.lm, train$AvgListing, val$AvgListing)
mean(abs(preds.lm - val$AvgListing)) #MAE
sqrt(mean((preds.lm - val$AvgListing)^2)) #RMSE

#boosting without average rental
mod.boost=gbm(AvgListing ~ . - ZipCode - AvgRental,data = train, distribution =
 "gaussian", n.trees = 1500,
                  shrinkage = 0.1, interaction.depth = 4)
mod.boost

summary(mod.boost)

predict.boost = predict(mod.boost, newdata = val, n.trees = 1500)

#boosting performance
OSR2(predict.boost, train$AvgListing, val$AvgListing)
mean(abs(predict.boost - val$AvgListing))
sqrt(mean((predict.boost - val$AvgListing)^2))

#Part 2

#Linear Regression with selected features
lin.mod <- lm(AvgListing ~ AvgRental + CA.AGI + Average.Household.size +
              Total.population + Population.Per.Square.Mile..Land.Area. +
              Vacancy.rate + Total.Asian + Total.NHOPI +
               Black.or.African.American, data = train)
summary(lin.mod)

preds.lm <- predict(lin.mod, newdata = test)

#linear regression test set MAE, RMSE, OSR2 (un-normalized)
OSR2(preds.lm, train$AvgListing, test$AvgListing)
mean(abs(preds.lm - test$AvgListing)) #MAE
sqrt(mean((preds.lm - test$AvgListing)^2)) #RMSE

#CART Regression with selected features
cart.mod <- rpart(AvgListing ~ AvgRental + CA.AGI + Average.Household.size +
              Total.population + Population.Per.Square.Mile..Land.Area. +
              Vacancy.rate + Total.Asian + Total.NHOPI +
               Black.or.African.American,
                  data = train, method = "anova", cp = 0.02, minsplit = 10)

CartPredictions <- predict(cart.mod, newdata=test)
```

```
#CART test set MAE, RMSE, OSR2 (un-normalized)
OSR2(CartPredictions, train$AvgListing, test$AvgListing)
mean(abs(CartPredictions - test$AvgListing))
sqrt(mean((CartPredictions - test$AvgListing)^2))

#cross-validated random forest with selected features
set.seed(311)
train.rf.b = train(AvgListing ~ AvgRental + CA.AGI + Average.Household.size +
              Total.population + Population.Per.Square.Mile..Land.Area. +
              Vacancy.rate + Total.Asian + Total.NHOPI +
               Black.or.African.American,
                 data = train,
                 method = "rf",
                 tuneGrid = data.frame(mtry = 1:9),
                 trControl = trainControl(method = "cv", number = 5,
                  verboseIter = TRUE))
train.rf.b
train.rf.b$results

mod.rf.b = train.rf.b$finalModel
predict.rf.b = predict(mod.rf.b, newdata = test)

#random forest performance
OSR2(predict.rf.b, train$AvgListing, test$AvgListing)
mean(abs(predict.rf.b - test$AvgListing))
sqrt(mean((predict.rf.b - test$AvgListing)^2))

#boosting with selected features
mod.boost=gbm(AvgListing ~ AvgRental + CA.AGI + Average.Household.size +
              Total.population + Population.Per.Square.Mile..Land.Area. +
              Vacancy.rate + Total.Asian + Total.NHOPI +
               Black.or.African.American,
              data = train, distribution = "gaussian", n.trees = 13000,
                  shrinkage = 0.001, interaction.depth = 8)
mod.boost

summary(mod.boost)

predict.boost = predict(mod.boost, newdata = test, n.trees = 13000)

#boosting performance
OSR2(predict.boost, train$AvgListing, test$AvgListing)
mean(abs(predict.boost - test$AvgListing))
sqrt(mean((predict.boost - test$AvgListing)^2))
```