Stacking: Universal Dataset

The SaratogaHouses dataset has 16 variables and 1728 records. Use “price” as target variable.

A data frame with 1728 observations on the following 16 variables.

- `price` price (1000s of US dollars)
- `lotSize` size of lot (square feet)
- `age` age of house (years)
- `landValue` value of land (1000s of US dollars)
- `livingArea` living are (square feet)
- `pctCollege` percent of neighborhood that graduated college
- `bedrooms` number of bedrooms
- `firplaces` number of fireplaces
- `bathrooms` number of bathrooms (half bathrooms have no shower or tub)
- `rooms` number of rooms
- `heating` type of heating system
- `fuel` fuel used for heating
- `sewer` type of sewer system
- `waterfront` whether property includes waterfront
- `newConstruction` whether the property is a new construction
- `centralAir` whether the house has central air

1. Import the data into R

```
  # Load the required libraries
        library(vegan)
        library(randomForest)
        library(infotheo)
        library(C50)
        library(rpart)
        library(dummies)
        library(e1071)
        library(DMwR)
        library(vegan)
# Set  working directory
         setwd("C:/Users/ashwin/Desktop/22nd Batch")
# Read the data from csv file
            data = read.csv(file = "SaratogaHouses.csv", header = TRUE, col.names = attr)
            str(data)
```

2. Convert the attributes to appropriate types and combine the numeric and categorical attributes.

```
    attr <-
        c('price','lotsize','age','landValue','livingArea','pctCollege','bedrooms','fireplaces','bathrooms','roo
        ms','heating','fuel','sewer','waterfront','newConstruction','centralAir')
     cat_Attr = c('heating','fuel','sewer','waterfront','newConstruction','centralAir')
     num_Attr = setdiff(attr, cat_Attr)
      num_Attr_without_target = setdiff(num_Attr,"price")
      cat_Data = data.frame(sapply(data[,cat_Attr], as.factor))
       num_Data = data.frame(sapply(data[,num_Attr], as.numeric))
      data <- cbind(num_Data,cat_Data)
```

3. Standardize the numeric data

```
    data[,num_Attr_without_target] <- decostand(data[,num_Attr_without_target],'range')
```

4. Convert all categorical attributes to numeric using the dummy function, then replace the old categorical attributes with their dummied values.

```
    heating <- dummy(data$heating)
     fuel <- dummy(data$fuel)
     sewer <- dummy(data$sewer)
```

```
waterfront <- dummy(data$waterfront)
newConstruction <- dummy(data$newConstruction)
centralAir <- dummy(data$centralAir)

data = subset(data,select= -c(heating,fuel,sewer,waterfront,newConstruction,centralAir))
data <- cbind(data,heating,fuel,sewer,waterfront,newConstruction,centralAir)
```

5. Changing certain column names

```
names(data)[c(12,13,18)]
names(data)[c(12,13,18)] <-
        c('heatinghot_air','heatinghot_water_steam','sewerpublic_commercial')
```

6. Divide the data into train and test

```
set.seed(1234)
train_RowIDs = sample(1:nrow(data), nrow(data)*0.7)
train_Data = data[train_RowIDs,]
test_Data = data[-train_RowIDs,]
```

7. Build several regression models

```
# Build CART model on the training dataset
cart_Model = rpart(price~., train_Data,method = "anova")
summary(cart_Model)

#Build a Random Forest model on the training dataset
rf_Model <- randomForest(price~., data=train_Data, ntree=50,keep.forest=T, importance=TRUE)
summary(rf_Model)

#Build S.V.M model for the training dataset
svm_model <- svm(train_Data[,-1],train_Data$price,type = "nu-regression")
summary(svm_model)
```

8. Predicting on train dataset

```
#---------Predict on Train Data----------
# Using CART Model predict on train data
cart_Train = predict(cart_Model, train_Data, type = "vector")
regr.eval(cart_Train,train_Data$price)

#Using random forest to predict data on train dataset
rf_Train <-predict(rf_Model,train_Data)
regr.eval(rf_Train,train_Data$price)

#Using S.V.M to predict training dataset
svm_Train <- predict(svm_model,train_Data[,-1])
regr.eval(svm_Train,train_Data$price)
```

9. Combining the training predictions of all the models.

```
train_Pred_All_Models = data.frame(CART = cart_Train,
                                   SVM = svm_Train,
                                   RandomForest = rf_Train)
```

10. Add the original target variable to the dataset.

```
# Adding the original DV to the dataframe
train_Pred_All_Models = cbind(train_Pred_All_Models, price = train_Data$price)
```

11. Ensemble the model with lm as Meta Learner

```
#Meta-learner model
ensemble_Model = lm(price ~ ., train_Pred_All_Models)
summary(ensemble_Model)
```

12. Evaluate the ensembled model on train data

```
ensemble_Train = predict(ensemble_Model, train_Pred_All_Models)
regr.eval(ensemble_Train,train_Data$price)
```

13. Follow the steps from 7 to 12 on the test data and evaluate the model.