Introduction to Data Science

Instructor: Daniel D. Gutierrez

ALTERNATE CLASS PROJECT

Supervised Machine Learning - Regression

ASSIGNMENT

This alternative class project is designed for you to become productive with your new data science skills by working with a real-life, California Housing data set. We'll use the data set to predict median house values. This data set appeared in a 1997 paper titled Sparse Spatial Autoregressions by Pace, R. Kelley and Ronald Barry, published in the *Statistics and Probability Letters* journal. The researchers built it using the 1990 California census data. It contains one row per census block group. A block group is the smallest geographical unit for which the U.S. Census Bureau publishes sample data (a block group typically has a population of 600 to 3,000 people). The variables found in the data set are as follows (the names are fairly self-explanatory):

`longitude`

`latitude`

`housing_median_age`

`total_rooms`

`total_bedrooms`

`population`

`households`

`median_income`

`median_house_value`

`ocean_proximity`

Each row pertains to a group of houses representing medians for groups of houses in close proximity.

The goal for this project is for you to gain experience in trying out the principles of data science using R as discussed in class. You'll need to carry out the tasks outlined below that parallel the data science process detailed in class.

1. Access the Data Set

The first step is to access the data set and load it into the R environment. Follow these steps in order to complete this step:

- Download the `housing.csv` data set found in the `homeworks` folder in the class GitHub repository. The data set has 20,640 rows and 10 variables.
- Read the data set into R using a data frame named `housing`.
- Perform `head()` and `tail()` functions on the data frame to get a feel for the actual data values.
- Perform a `summary()` function on the data frame to get a sense for the data classes, range of values for numeric variables, and levels for any factor variable.

2. Data Visualization

Now, let's do some exploratory data analysis (EDA) using some visualizations. Follow these steps:

- Create histograms for each numeric variable (e.g. not `ocean_proximity`).
- Examine the plots and provide a commentary on what the visualizations reveal.

3. Data Transformation

The next step is to transform data as necessary. Follow these steps in order to complete this step:

- We see from the `summary()` results above that there are many NA values in the `total_bedrooms` variable (the only variable with missing values). This needs to be addressed by filling in missing values using imputation. You can use the "median" for missing `total_bedrooms` values. The median is used instead of mean because it is less influenced by extreme outliers. This may not be the best method, as these missing values could represent actual buildings (e.g. a warehouse) with no bedrooms, but imputation often makes the best of a bad situation.
- Split the `ocean_proximity` variable into a number of binary categorical variables. Many machine learning algorithms in R can handle categoricals in a single column as a factor class, but we will cater to the lowest common denominator and do the splitting. Once you're done splitting, you can remove the `ocean_proximity` variable.
- Use the `total_bedrooms` and `total_rooms` variables (and `households`) to create new `mean_number_bedrooms` and `mean_number_rooms` variables as these are likely more accurate depictions of the houses in a given group. You can then remove the `total_bedrooms` and `total_rooms` variables.
- Perform feature scaling. Scale each numerical variable except for `median_house_value` (as this is what we will work to predict). The predictor values are scaled so that coefficients in some machine learning algorithms are given equal weight.
- The result of your data transformation processes, you should have a new data frame named `cleaned_housing` with the following variables:

```
"NEAR BAY"             "<1H OCEAN"            "INLAND"
"NEAR OCEAN"           "ISLAND"               "longitude"
"latitude"             "housing_median_age"   "population"
"households"           "median_income"        "mean_bedrooms"
"mean_rooms"           "median_house_value"
```

4. Create Training and Test Sets

Now we can prepare for machine learning by creating the training and test sets using a random sample index.

- Create a random sample index for the `cleaned_housing` data frame.
- Create a training set named `train` consisting of 80% of the rows of the `housing` data frame.
- Create a test set named `test` consisting of 20% of the rows of the `housing` data frame.

5. Supervised Machine Learning - Regression

In this step, you'll use the `randomForest()` algorithm found in the `randomForest` package for training and inference. Our goal is to predict the median house value.

First, you'll need to separate your training set `train` into two pieces: `train_x` and `train_y` where `train_x` shall have all variables <u>except</u> the response variable `median_house_value` and train_y shall have <u>only</u> the response variable `median_house_value`.

Next, you'll call the `randomForest()` algorithm, passing to it both components of the training set created above. Make sure you specify `ntree=500`, and `importance=TRUE`. Return the resulting model in the variable `rf`.

```
rf = randomForest(x=train_x, y=train_y ,
                                ntree=500, importance=TRUE)
```

Now use `names(rf)` to see all the different metrics computed by the algorithm. Display `rf$importance` to see Mean Squared Error (MSE) which is a measure of feature importance. It is defined as the measure of the increase in MSE of predictions when the given variable is shuffled, thereby acting as a metric of that given variable's importance in the performance of the model. So a higher number indicates a more important predictor.

6. Evaluating Model Performance

With the random forest algorithm, there is no need for a separate test set to get an unbiased estimate of the test set error. It is estimated internally, during the run, as follows: each tree is constructed using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of the kth tree. Compute the out-of-bag (oob) error estimate using:

```
# Not specifying a data source forces OOB predictions
oob_prediction = predict(rf)

# Now compute the training set RMSE
train_mse = mean(as.numeric((oob_prediction -
train_y)^2))
oob_rmse = sqrt(train_mse)
oob_rmse
```

The resulting RMSE is your prediction of median price of a house in a given district to within a RMSE delta of the actual median house price. This can serve as your benchmark moving forward and trying other statistical models. Next, we can see how well the model predicts using the test data.

Split the test set in the same manner as the training set above, creating two new data frames: `test_x` and `test_y`. Use this code to make the predictions:

```
y_pred = predict(rf , test_x)

# Now compute the test set RSME
test_mse = mean(((y_pred - test_y)^2))
test_rmse = sqrt(test_mse)
test_rmse
```

How does the test set RMSE compare with the training set RMSE? Did the model score roughly the same on the training and testing data, suggesting that it is not overfit and that it makes good predictions?

Congratulations! You've just gone through the entire data science process to create a machine learning model to predict the median housing value. As with the homework assignments, please upload single R script file to Canvas containing all your project code including comment lines that include the output of your R code.