Máster Oficial Universitario en
Ingeniería Informática
muiinf.webs.upv.es

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA
etsinf

Data Science (CDA)

Practical 6:

Regression Models

José Hernández Orallo (jorallo@dsic.upv.es)
(Adapted from material by M. José Ramírez Quintana)

In this session we are going to illustrate prediction models by using several regression techniques seen in Unit 3.

A regression case study

We are going to work with data about house prices in USA for year 2013 downloaded from Zillow (a real-estate agency in USA) API (http://www.zillow.com/howto/api/APIOverview.htm). The data (available at poliformat on a csv file) has no missing values.
The Zillow dataset contains features describing the houses (number of bathrooms, total number of rooms, number of bedrooms, area, year of construction), the house address (zip code, street, city, state) and the price.

Exercises

1. Load the dataset into R and remove the street information in order to anonymise the data. Explore the data with *str* and *summary*.

2. Print a scatter plot of price versus each other variable so that you can see which variables are likely to be most important. This can be done by creating a data frame for plotting, using the *make.groups* function from the *lattice package*. Attach the data frame to the search path (by using the *attach* function) to refer to the variables in the data frame by their names alone. Then, the new data frame is created as follows:

forplot<-make.groups(bath=data.frame(value=bath,zipcode,city,state,price),
year=data.frame(value=year,zipcode,city,state,price),
bed=data.frame(value=bed,zipcode,city,state,price),
rooms=data.frame(value=rooms,zipcode,city,state,price),
SqFt=data.frame(value=SqFt,zipcode,city,state,price))
And use the *detach* function (to remove the data frame *houses* from the search path).
Finally, the scatter plot is drawn with the xyplot function as follows:
xyplot(price~value|which, data=forplot,scales=list(relation="free"))
Do you see any interesting relationship between the price and the other variables?

3. Randomly split the dataset into 75% train and 25% test.

4. Fit several regression models to the training data but only using the numerical attributes: a linear model (using the *lm* function, which fits a linear model using ordinary least squares), a regression tree (*CART*) from the *rpart* package (set the parameter *method* to *anova* in order to produce a CART tree), and a neural network from the *nnet* package (set the parameters *skip* and *linout*-numerical output- to *TRUE* and *size*-hidden units- to 12).

5. View the models using the *summary* method and, additionally, the *plot* method for the CART tree. Which one is the less informative?

6. Prune the regression tree using the *prune* function and setting the *cp* parameter to the value you consider is a good balance between complexity and performance (the

plotcp function plots tree sizes and relative errors for different values of the complexity parameter). Visualise the new tree.

7. Compare how each method works using the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) for the training and test data sets. Which model performs better for the training data? And for the test data?

8. Can you improve the results by changing the parameters or trying other methods?