Week 3 Topics

1. Chapter 6 – Regression

# The Difference between Regression and Classification (in machine learning context)

Although Classification and Regression come under the same umbrella of Supervised Machine Learning and share the common concept of using past data to make predictions, or take decisions, that’s where their similarity ends.

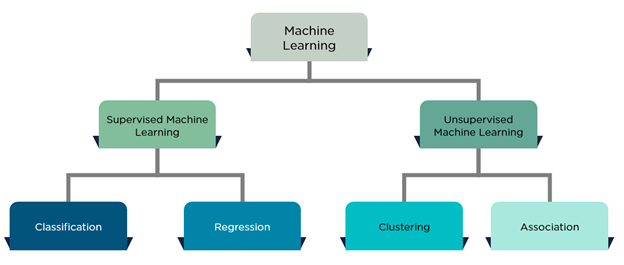


Figure 1: Machine Learning Algorithms Class

Classification problems, requires items to be divided into different categories, based on past data. In a way, we’re solving a yes/no problem. Whether something meets its required standards, or whether it’s faile or not and so on.

Now with regression problem, the system attempts to predict a value for an input based on past data. Unlike classification, we are predicting a value based on past data, rather than classifying them into different categories.

Regression is used to predict continuous values. Classification is used to predict which class a data point is part of (discrete value).

## **Simple Linear Regression**

In statistics, simple linear regression is the least squares estimator of a linear regression model with a single explanatory variable. In other words, simple linear regression fits a straight line through the set of n points in such a way that makes the sum of squared residuals of the model (that is, vertical distances between the points of the data set and the fitted line) as small as possible.

The adjective simple refers to the fact that the outcome variable is related to a single predictor. The slope of the fitted line is equal to the correlation between y and x corrected by the ratio of standard deviations of these variables. The intercept of the fitted line is such that it passes through the center of mass (x, y) of the data points.

**Linear function**

In calculus and related areas of mathematics, a linear function from the real numbers (*X*) to the real numbers (*Y*) is a function whose graph (in Cartesian coordinates with uniform scales) is a line in the plane. The characteristic property of linear functions is that when the input variable is (*X*) changed, the change in the output (*Y*) is proportional to the change in the input.

Linear functions are related to linear equations. The linear function is represented by:

Where .

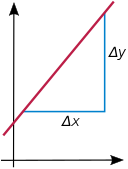


Figure 2: Y-intercept and slop of a line

Now, let’s assume there is a two column data sets, column of Xs and Column of Ys.

Where Xs = {*x1, x2, x3, …, xi, …, x*n} and Ys = { *y1, y2, y3, …, yi, …, y*n}

Thus, given a dataset of Xs and Ys and **assuming**, these are correlated and best function representing their relationship is a linear function then, we can write .

The is the error imbedded in our assumption. Because for a the value of doesn’t reside on the line of with an error of

Now, let’s assume *x*i is predicting (Predictor) the value of yi with an error . That is we have only the value of and we want to find (predict) the value of . So we can use the line function using the original given dataset and predict a new value for :

Where are some constant (intercept and slop) and is the prediction error. As we see the relationship between Target value and Predictor is a liner equation. That is, we use regression only when there is a linear relationship between the Target and Predictor.

Notation:

,the *ith* independent value

, the *ith* dependent value

Given a random sample from the population (in analytics most of the time we use the entire population instead of a random sample), we estimate the population parameters and obtain the sample linear regression model:

The residual,, ei=yi−y^iis the difference between the value of the dependent variable predicted by the model, y^i, and the true (actual) value of the dependent variable, yi. One method of estimation is ordinary least squares. This method obtains parameters’ estimates that minimize the Root of Sum of Squared Residuals or Root of Sum of Squared Errors RSSE:

or

## **Quality of Regression Model**

The goal is to find equation of the straight line;

where is the estimated , is the estimated Y-intercept, and is the estimated slope, which would provide a "best" fit for the data points. Here the "best" will be understood as in the ***least-squares*** approach: a line that minimizes the sum of squared residuals of the linear regression model. In other words,  (the *Y*-intercept) and  (the slope) solve the following minimization problem.

Find , for . Obviously Q is our objective function. We can use linear programming technique to solve this problem.

To calculated the value of estimated , , such that the value is minimum we use the following method.

Where

And

Where and ( are the **mean** of Xs and Ys, that is )

It is important to notice that considering no regression present, then

.

This value also represent the estimated error in respect to the mean of Ys line (no regression).

If we use the training data thus obviously we have all the (actual) values then, the

, changes to . To calculate the total errors and to treat negative and positive errors, we calculate the error as the Sum of Squared Errors

In the case the y values are predicted then:

Considering the same approach when we have a regression.

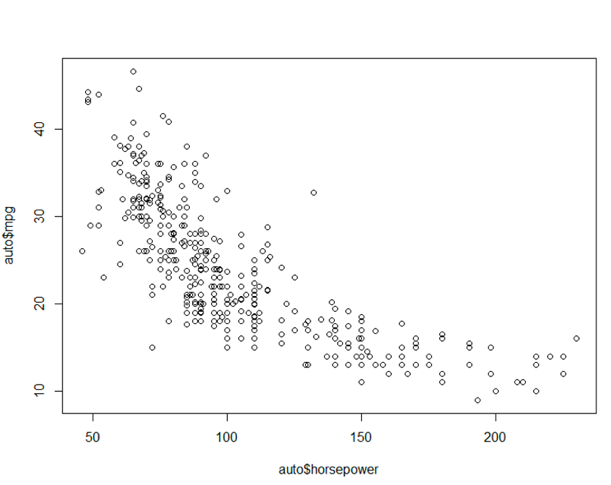
## **Simple Linear Regression with R**

SLR example : Dataset is posted in week 3 folder on Canvas

auto <- read.csv(“auto-mpg.csv”)

View data: plot mpg vs horsepower

plot(“auto$horsepower, auto$mpg)



The plot shows a negative and not linear relationship. But since we are interested in the linear we proceed with the work. If you are interested with nonlinear, then try Excel trendline.

Step one: Partitioning data into training and test.

We have to load the caret package. Library(“caret”)

set.seed(2015)

sam <-createDataPartition(auto$mpg, p = 0.7, list = FALSE) where mpg is the target variable.

Train <- auto[sam,] and test<- auto[-sam,]

Step two: Build the model and verify the result

Using training partition to build our regression model. We use R function ***lm*** implement linear regression. Specifying target and predictor variables.

The regression function is:

Auto.lm[“coeficients”]

Intercept = 39.5768770

Predictor coefficient = -0.1543497

mpg = 39.5768770 -0.1543497horsepower

we can get the process summary as follow

summary(auto.lm)

we can plot the line and see how it fits the plotted data (just the training)

you need to create plot of data first and then paste the regression line on it.

plot(train$horsepower, train$mpg, xlab = “Horsepower”, ylab = “Mile/Galon”)

abline(auto.lm)

Step three : Apply the model on the test data, and verify the quality

Since we have the regression function from the training data, we can use it on the test data to predict the values of target (mpg) and compare them with the actual mpg values

auto$Pred.test <- predict(auto.lm, test)

When the result of the applying model is available we can put it in a table of two columns with the actual mpg values in the test data. Calculating RMSE with R

evaltable <- data.frame(test$mpg, pred.test)

rmse <- sqrt(mean((test$mpg, pred.test)2))

rmse

the result will be 5.486

## **Multiple Linear Regression**

Multiple linear regression is an expansion of a simple linear regression into multiple predictors regression. it attempts to model the relationship between two or more predictor attributes (or variables) and a target attribute by fitting a linear equation to observed data. Every value of the independent variable X is associated with a value of the dependent variable y. The population regression line for *p* predictor attributes is defined by:

The assumption is that the above function approximates the relationship between the predictors and target. Since the observed values for *Y* vary about their means *Y* the multiple linear regression model includes a term for this variation. In words, the model is expressed as DATA = FIT + RESIDUAL, where the "FIT" term represents the expression:

The "RESIDUAL" term represents the deviations of the observed values *Yi* from their means Y, which are normally distributed with mean µ and variance σ. The notation for the model deviations is

Since every predictors has many values then we can write, for example for predictor *X1*

Considering having n observations.

In the least-squares model, the best-fitting line for the observed data is calculated by minimizing the sum of the squares of the vertical deviations from each data point to the line (if a point lies on the fitted line exactly, then its vertical deviation is 0). Because the deviations are first squared, then summed, there are no cancellations between positive and negative values.

The outcome of the multiple linear regression (target\_atrribute) is denoted by and the residual , where the index *i* denotes the *ith* values of all *p* predictors.

Regression modeling means not only estimating the coefficients but also choosing which predictors to include and what form. For example, numerical predictors can be included as is, or in logarithmic form [lob(X)], or in a binned form (e.g. age group). Choosing the right form depends on domain knowledge, data availability, and needed predictive power.

## **Explanatory vs. Predictive Modeling**

We already discussed the linear regression formula in this paper then, we should know the important distinction which, often escapes those with limited familiarity with linear regression from courses in statistics. In particular, the two popular but different objectives behind fitting a regression mode are:

* Exploratory; Explaining or quantifying the average effect of inputs on an outcome (explanatory or descriptive task, respectively)
* Predicting the outcome value for new records, given their input values (predictive task)

The statistical approach is focused on the first objective. In that scenario, the data are treated as random sample from a larger population of interest. The regression model estimated from this sample is an attempt to capture the average relationship in the larger population. This model then it is used in decision making. For example using the regression to explore the sensitivity of output based on changes in one or more input considering all other attributes unchanged.

In predictive analysis, however, the focus is typically on the second goal: predicting a new records. Here we are not interested in the coefficients themselves, nor in the “average record.” But rather in the predictions that this model can generate for new records.

However, both approach are similar in all steps excepts two phase usually leading to the final models. Therefore, the choice of the model is closely tied to whether the goal is exploratory or predictive. The main differences in using a linear regression in the two scenarios:

1. A good exploratory model is the one fits the data closely, whereas a good predictive model is one that predict new record accurately. Thus, choices of input variables and their form can differ.
2. In exploratory model, the entire dataset is used for estimating the best fit model, to maximize the amount of information that we have about the hypothesized relationship in the population. When the goal is to predict outcomes of the new individual records, data is partitioned, training set is used for building the regression model and validation set is used for assess the model.
3. Performance measures for exploratory models measure how close the data fit the model (how well the model approximates the data) and how strong the average relationship is, but in predictive models performance is measured by predictive accuracy .
4. In exploratory models the focus is on the coefficients (, but in predictive models the focus in on predicting ()

## **Multiple Linear Regression with R**

We are going to use the education.csv to show how a multiple linear regression (MLR) is built and evaluated in R. This dataset has 6 attributes. The “expense” is the target attribute. The rest of them are predictors. “state” and “region” are categorical attributes. We will ignore state and transform the region into numerical.

1. **Reading in the dataset**

ed<-read.csv("education.csv") #Dataset is posted in week 3 folder on Canvas

1. **Check the attributes to see there is any outliers (just for under18) (see figure below)**

boxplot(ed$under18)

outliers<-boxplot(ed$under18, plot = F)$out

ednew<-ed[!(ed$under18 %in% outliers),]

1. **Change a numeric attribute to categorical (region in this case)**

ednew$region <-factor(ednew$region)

1. **Partition the dataset**

set.seed(2015)

sam<-createDataPartition(ednew$expense, p=0.7, list = FALSE)

train<-ednew[sam,]

test<-ednew[-sam,]

1. **Build the MLR model on training dataset**

ed.lm<-lm(train$expense ~ ., data = train[,-1])

1. **Display the outcome for analysis**

lm(formula = train$expense ~ ., data = train[, -1])

Residuals:

Min 1Q Median 3Q Max

-65.762 -17.714 -1.467 17.335 67.428

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -130.02750 160.81803 -0.809 0.4254

region2 -12.18171 20.81608 -0.585 0.5629

region3 -5.57296 18.03390 -0.309 0.7595

region4 22.83198 19.31246 1.182 0.2467

urban 0.06266 0.06111 1.025 0.3137

income 0.04618 0.01499 3.080 0.0045 \*\*

under18 0.46407 0.41266 1.125 0.2700

---

Signif. codes: 0 ë\*\*\*í 0.001 ë\*\*í 0.01 ë\*í 0.05 ë.í 0.1 ë í 1

Residual standard error: 33.94 on 29 degrees of freedom

Multiple R-squared: 0.6067, Adjusted R-squared: 0.5254

F-statistic: 7.457 on 6 and 29 DF, p-value: 6.791e-05

As we see among all predictors *income* effect is significant, thus we rebuild the model with only the *income* predictor.

1. **Build the MLR with only income predictor**

ed.lm.1<-lm(train$expense ~ income, data = train[, -1])

1. **Analyze the result**

summary(ed.lm.1)

Call:

lm(formula = train$expense ~ income, data = train[, -1])

Residuals:

Min 1Q Median 3Q Max

-75.31 -24.03 -2.90 18.51 83.48

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 44.124594 43.371450 1.017 0.316

income 0.050317 0.009231 5.451 4.46e-06 \*\*\*

---

Signif. codes: 0 ë\*\*\*í 0.001 ë\*\*í 0.01 ë\*í 0.05 ë.í 0.1 ë í 1

Residual standard error: 36.52 on 34 degrees of freedom

Multiple R-squared: 0.4663, Adjusted R-squared: 0.4506

F-statistic: 29.71 on 1 and 34 DF, p-value: 4.459e-06

1. **Evaluate the model on the test dataset and output the final work in a csv file.**

RMSE<-sqrt(mean((test$expense-pred.test)^2))

RMSE

[1] 41.20956

PredictedED<-data.frame(test, pred.test)

write.csv(PredictedED, file = "Predicted-education.csv")

1. **Create a visualization**



## **MLR Output Explained**

### Residuals

Normally it gives a basic idea about difference between the observed value of the dependent variable (Y) and the predicted value (X), it gives specific detail i.e. minimum, first quarter, median, third quarter and max value, normally it does not used in our analysis

### Coefficients-Intercept

We can see a all the remaining variable comes with one more row ‘Intercept’, Intercept is giving data when all the variables are 0 so all the measure done without considering any variable, this is again not much used in normal cases, it’s average value of y when x = 0

### Coefficient-Estimate

This is a one unit increase in X then expected change in Y.

### Coefficient-Std. Error

The standard deviation of an estimate is called the standard error. The standard error of the coefficient measures how precisely the model estimates the coefficient’s unknown value. The standard error of the coefficient is always positive.

Low value of this error will be helpful for our analysis, also used for checking confidence interval

### Coefficient-t value

t value = estimate/std error

high t value will be helpful for our analysis as this would indicate we could reject the null hypothesis, it is using to calculate p value

### Coefficient Pr(>|t|)

individual p value for each parameter to accept or reject null hypothesis, this is statistical estimate of x and y. Lower the p value allow us to reject null hypothesis. all type of errors (true positive/negative, false positive/negative) are come to picture if we wrongly analysis p value.

Asterisks mark aside p value define significance of value, lower the value have high significance

### Residual standard error

Residual standard error: 33.94 on 29 degrees of freedom (in final model of the education MLR model).

In normal work, average error of a model, how well our model is doing to predict the data on average

Degree of freedom is like no of data point taken in consideration for estimation taking parameter in account.

### Multiple R-squared and Adjusted R-squared

In our model in education dataset example,

Multiple R-squared: 0.6067, Adjusted R-squared: 0.5254

It is always between 0 to 1, high value are better Percentage of variation in the response variable that is explained by variation in the explanatory variable, this is used to calculate how well the model is doing to explain the things, when we increase no of variable then it will also increase and there are no proper limit to define how much we can increase.

We are taking dusted value in which we does not take all variables, only significant variable are considered in adjusted R squared

### F-statistic

In our model in education dataset example, F-statistic: 7.457 on 6 and 29 DF.

This is showing relationship between predictor and target, higher the value will give more reasons to reject null hypothesis, its significant of overall model not any specific parameter

### p-value

In our model in education dataset example, p-value < 6.791e-05

Overall p value on the basis of F-statistic, normally p value less than 0.05 indicate that overall model is significant