Matters of Discussion[ML]

Linear regression

- Simple & Multiple linear regression
- Estimating the regression equation
- prediction variable selection in linear regression

ML in Simple Linear Regression

- Simple linear regression is what you can use when you have one independent variable[**X**] and one dependent variable [**Y**].

$$Y = \beta_0 + \beta_1 X$$

- While training the model :
- x: input training data (univariate – one input variable(parameter))
- y: labels to data (supervised learning)
- **e.g.-For Given Height(X), predict the Weight (Y);**

Refer ---Linear Regression in Weka

ML in Multiple linear regression

- Multiple linear regression is what you can use when you have a bunch of different independent variables or predictor variables [X1,X2,..Xn] and one dependent variable or response variable[Y].
- The multiple linear regression explains the relationship between one continuous dependent variable (y) and two or more independent variables (x1, x2, x3... etc).

Cont..

- Multiple Input Factors and One predicted O/P.
- Dimension reduction technique may be used to filter out the good set of independent variables [Input Factors]that influence on the dependent variable[Y].

✓ Predictive variable Y or O/P .

- **Eye disease type prediction[predicted O/P] based on input disease symptoms [input factors].**

Refer ---Logistic Regression in Weka

Regression problem in machine learning

- A regression can have real valued or discrete input variables.
- Regression models are used to predict a continuous value.
- Predicting prices of a house given the features of house like size, price etc. is one of the common examples of Regression.

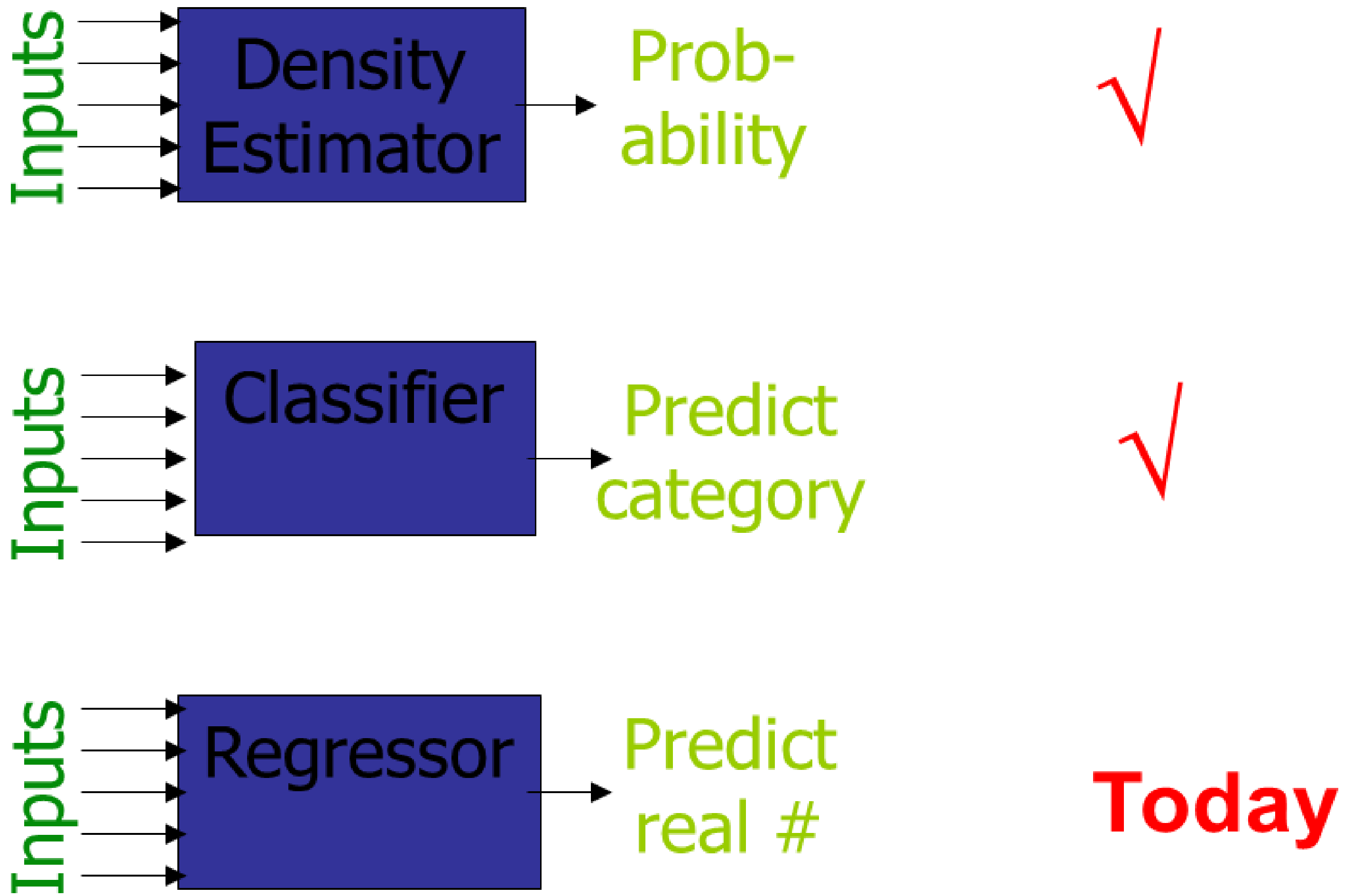
Continuous variables are simply running numbers.

Categorical variables are categories.

categorical: Well, I'm tall and smart

continuous: Well, I'm 180 cm and have an IQ of 126

Cont..


Inputs
Density Estimator
Prob-ability
√
Inputs
Classifier
Predict category
√
Inputs
Regressor
Predict real #
Today

Regression examples

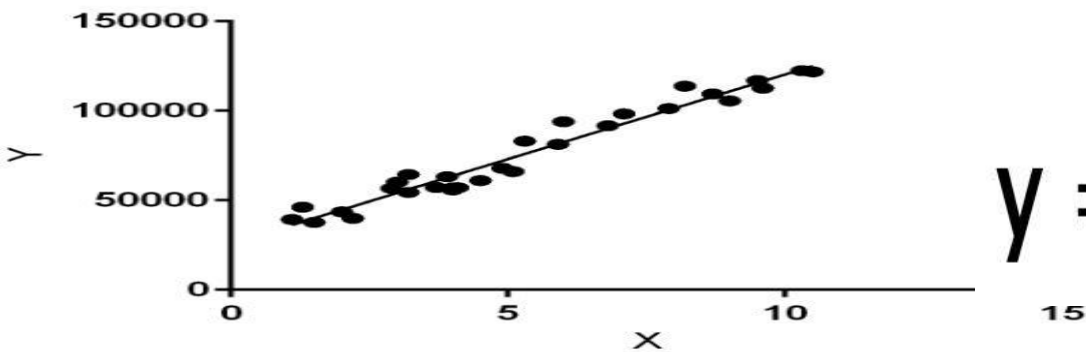
Stock market

Weather prediction

Temperature
72° F

Predict the temperature at any given location

Hypothesis function for Linear Regression :



$$y = \theta_1 + \theta_2 . x$$

When training the model –

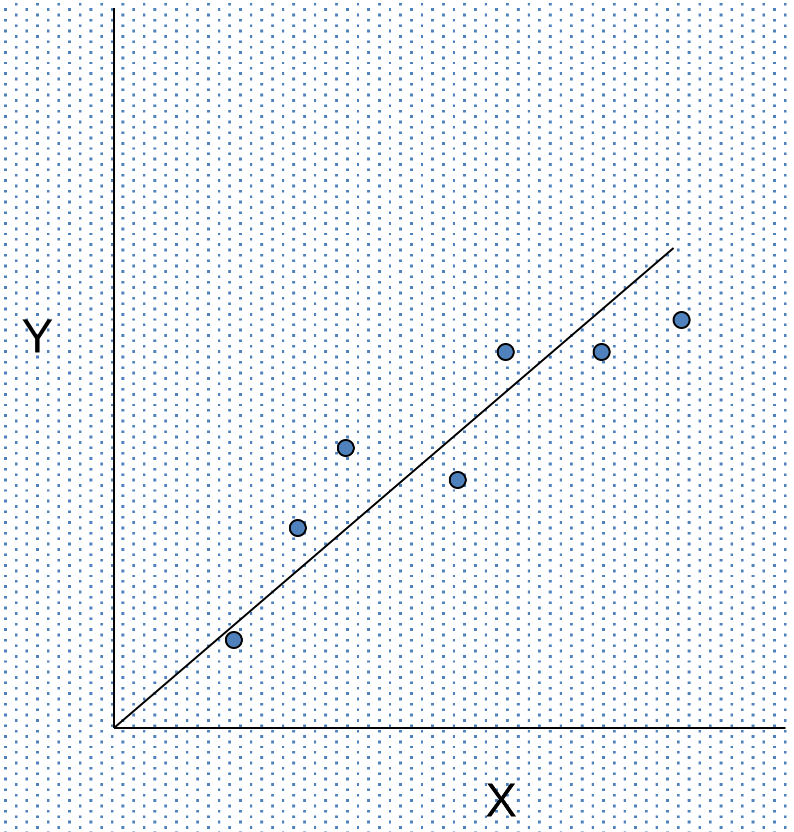
- it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.
  - θ1: intercept
  - θ2: coefficient of x
- **Once we find the best θ1 and θ2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.**

Linear regression Applications

❖ Given an input x we would like to compute an output y

❖ **For example:**

- Predict height from age
- Predict Google’s price from Yahoo’s price
- Predict distance from wall from sensors



Linear regression in ML

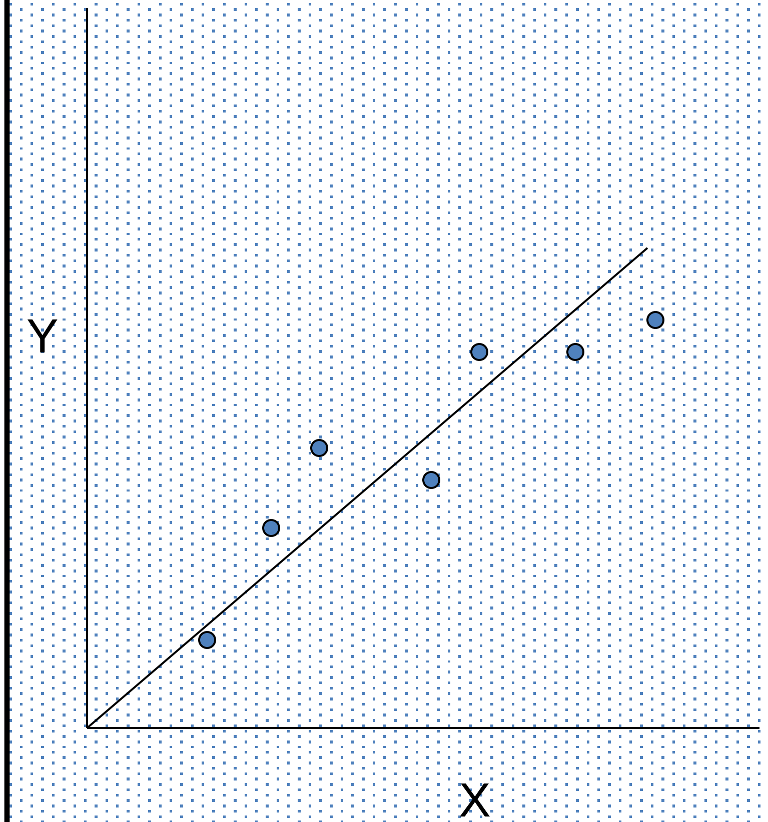
- Given an input x we would like to compute an output y
- In linear regression we assume that y and x are related with the following equation:

What we are trying to predict

Observed values

$$y = wx + \varepsilon$$

where w is a parameter and ε represents measurement or other noise



Linear regression in ML

- **Our goal is to estimate *w* from a training data of <x_i,y_i> pairs**

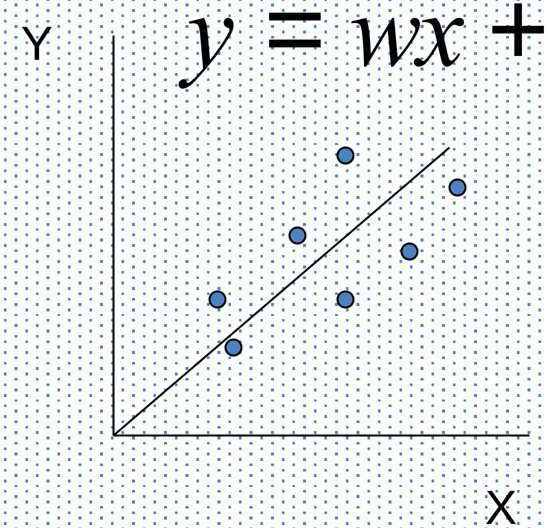
- Optimization goal: minimize squared error (least squares):

$$\arg\min_w \sum_i (y_i - wx_i)^2$$

- **Why least squares?**

  - minimizes squared distance between measurements and predicted line.

$$y = wx + e$$



Solving linear regression

- **To optimize:**
- We just take the derivative w.r.t. to w ….

prediction

$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i)$$

Training data of <xi,yi> pairs

Solving linear regression

- To optimize – closed form:
- We just take the derivative w.r.t. to w and set to 0:

$$\frac{\partial}{\partial w}\sum_i (y_i - wx_i)^2 = 2\sum_i -x_i(y_i - wx_i) \Rightarrow$$

$$2\sum_i x_i(y_i - wx_i) = 0 \quad \Rightarrow \quad 2\sum_i x_i y_i - 2\sum_i wx_i x_i = 0$$

Training data of <xi,yi> pairs

$$\sum_i x_i y_i = \sum_i wx_i^2 \quad \Rightarrow$$

$$w = \frac{\sum_i x_i y_i}{\sum x_i^2}$$

Implementation logic

```
# observations
    x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
    y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])


    # estimating coefficients
    b = estimate_coef(x, y)

    print("Estimated coefficients:\nb_0 = {Ɛ}  \
        \nb_1 = {w}".format(b[0], b[1]))
```

Analysis

$y = wx + \varepsilon$

Estimated coefficients:

ε = -0.0586206896552

w = 1.45747126437

The linear regression model ----------

y = **1.45747126437** x + (**-0.0586206896552**)

Cost function(J) of Linear Regression Model

Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (predicted) and true y value (y).

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

Reduced Cost function (minimizing RMSE value)

Best value of w and ε should be estimated that minimize the error between predicted y value (predicted) and true y value (y).

Steps to Establish a Regression

1) Carry out the experiment of gathering a sample of observed values of height and corresponding weight.
2) Create a relationship model using the lm() functions in R.
3) Find the coefficients from the model created and create the mathematical equation using these
4) To predict the weight of new persons, use the predict() function in R.

1. Input Data

Values of height

151, 174, 138, 186, 128, 136, 179, 163, 152, 131

Values of weight.

63, 81, 56, 91, 47, 57, 76, 72, 62, 48

2. lm() Function - creates the relationship model between the predictor and the response variable.

lm(formula,data)

formula is a symbol presenting the relation between x and y.

data is the vector on which the formula will be applied.

Cont..

Create Relationship Model & get the Coefficients

```
# Apply the lm() function.
relation <- lm(y~x)
print(relation)
```

N.B- "x ~ y" meaning that x and y are of the same order of magnitude.

3. O/P----Coefficients:

Find the coefficients from the model created and create the mathematical equation using these

```
(Intercept)          x
  -38.4551       0.6746
```

Y = 0.6746 x + (-38.4551)

4. use the predict() function in R.

predict(object, newdata)

object is the formula which is already created using the lm() function.

newdata is the vector containing the new value for predictor variable.

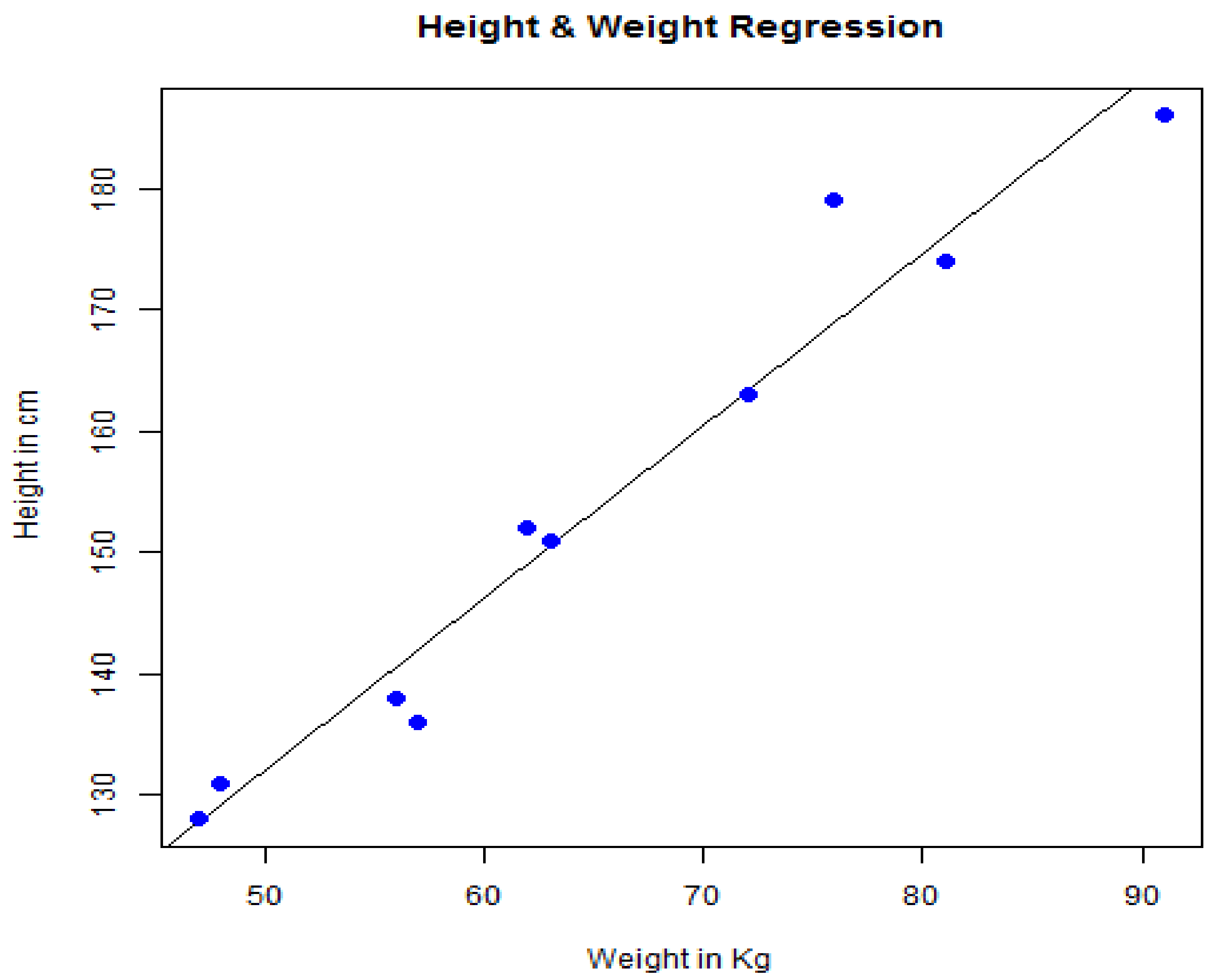
Predict the weight of new persons

```
# The predictor vector.

x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

# The resposne vector.

y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

# Apply the lm() function.

relation <- lm(y~x)

# Find weight of a person with height 170.

a <- data.frame(x = 170)

result <-  predict(relation,a)

print(result)
```

O/P-- 76.22869

```
# Create the predictor and response variable.
x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
relation <- lm(y~x)
# Give the chart file a name.
png(file = "linearregression.png")
# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression",
abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab =
"Height in cm")
```

https://www.tutorialspoint.com/r/r_linear_regression.htm


Height & Weight Regression
Height in cm
130
140
150
160
170
180
50
60
70
80
90
Weight in Kg

ACTIVITY -7(LAB-03)

Consider any dataset and Implement and investigate the Linear regression algorithm, and analyze the results in details.

ACTIVITY-08

Formulate Hypothesis function for Linear Regression and Investigate the computational analysis of linear regression model to estimate the coefficients for any real world application.

Cheers For the Great Patience!

Query Please?