Multiple Linear Regression

Short note about Linear Regression:

Multiple linear regressions are the most common form of the regression analysis. As a predictive analysis, multiple linear regressions are used to describe data and to explain the relationship between one dependent variable and two or more independent variables.

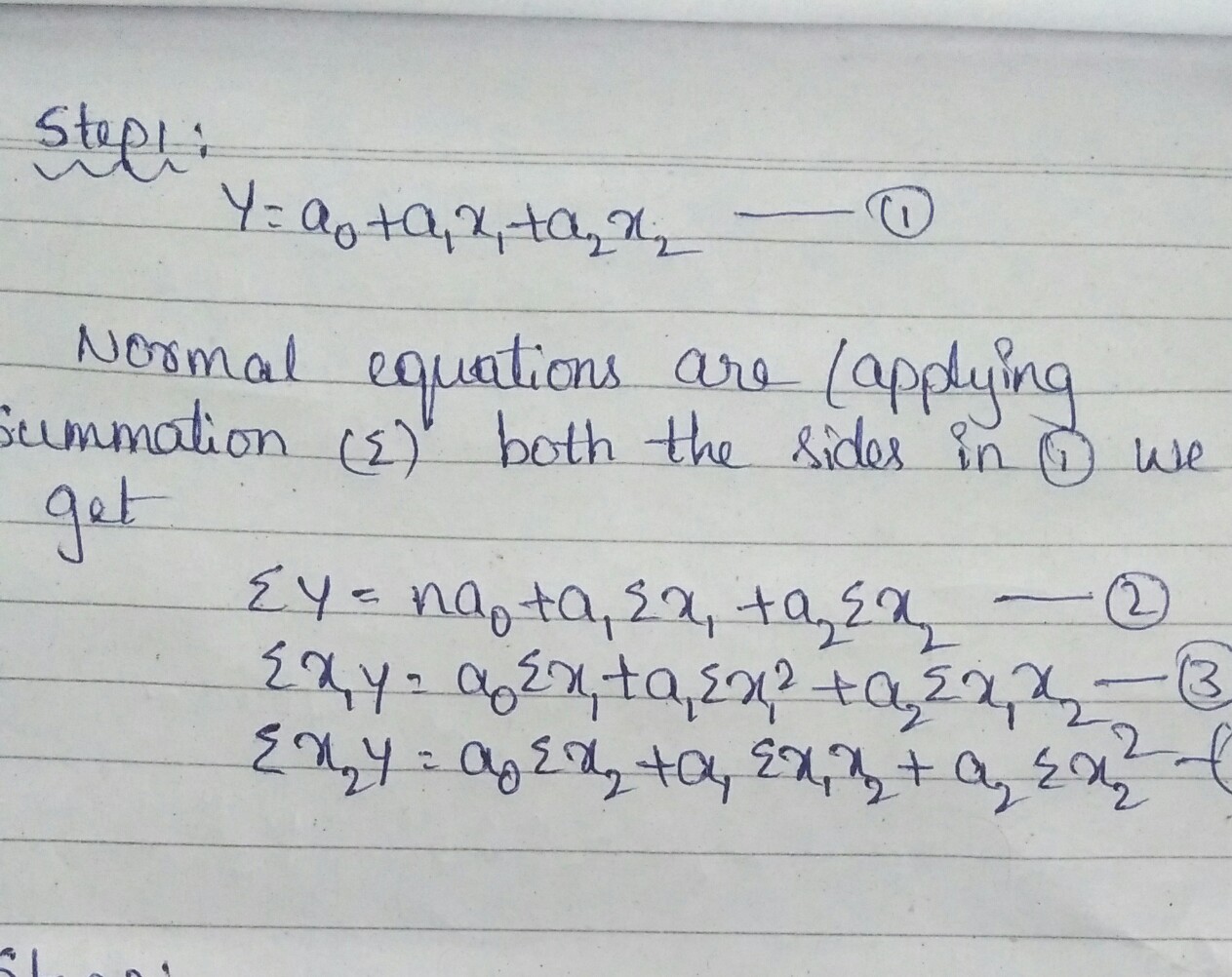
Decode Complex Algorithm

Step: 1

Y=a0+a1x1+a2x2

Step: 2

Normal Equation are -



Step: 2

|  |  |  |
| --- | --- | --- |
| **x1** | **x2** | **y** |
| 1 | 10 | 12 |
| 2 | 1 | 18 |
| 3 | 2 | 24 |
| 4 | 3 | 30 |

Considered y=Roll

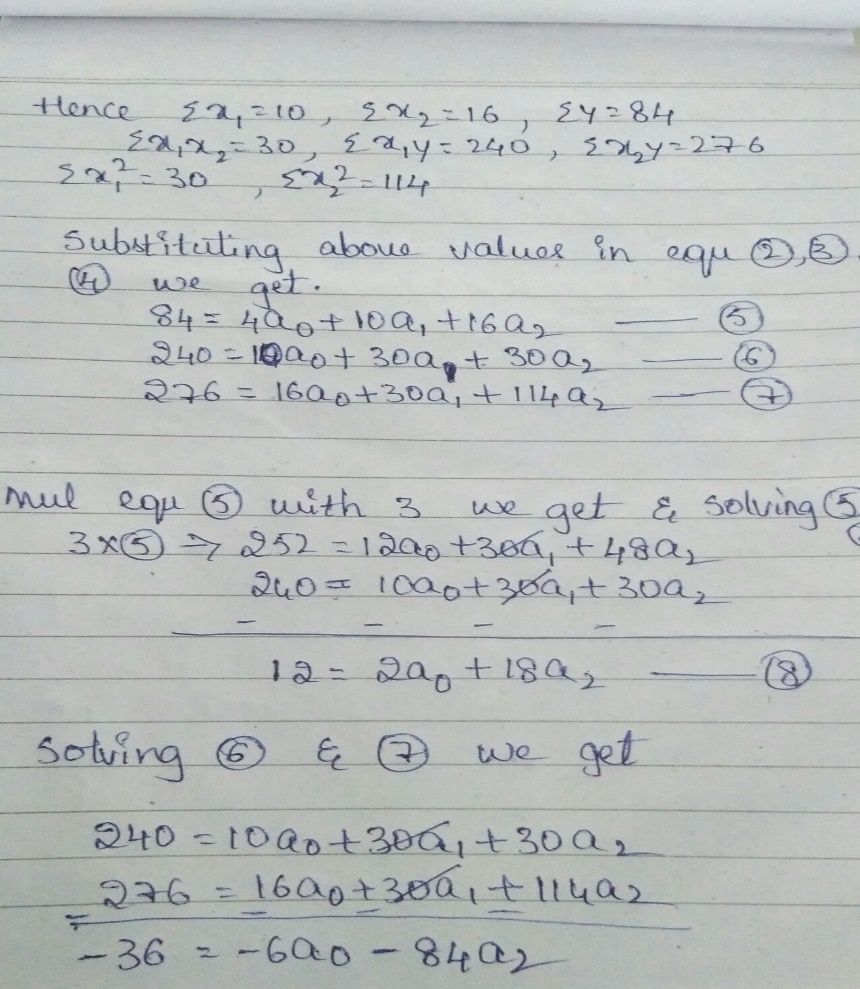
X1=Unemployee

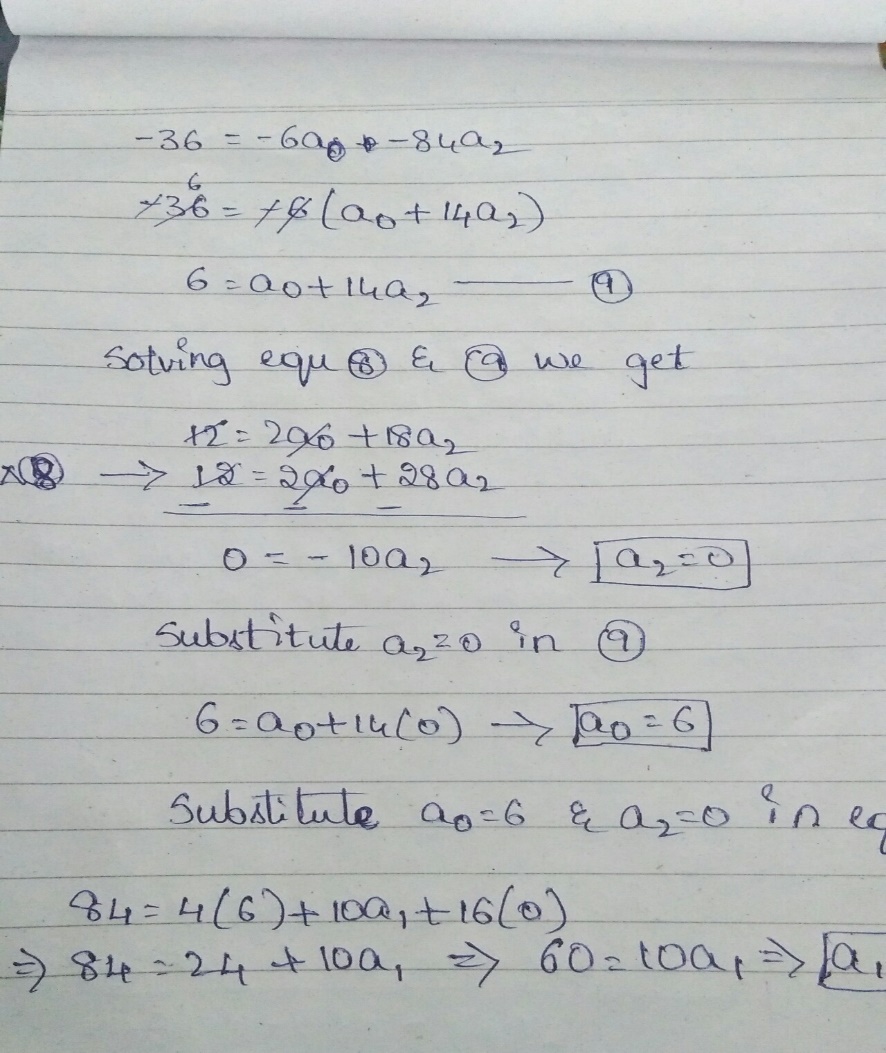
X2=Hupgrade

Solving B1 & B2 [ EXAMPLE: 1]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **x1** | **x2** | **y** | **x1x2** | **x1y** | **x2y** | **x1^2** | **x2^2** |
|  | 1 | 10 | 12 | 10 | 12 | 120 | 1 | 100 |
|  | 2 | 1 | 18 | 2 | 36 | 18 | 4 | 1 |
|  | 3 | 2 | 24 | 6 | 72 | 48 | 9 | 4 |
|  | 4 | 3 | 30 | 12 | 120 | 90 | 16 | 9 |
| Total | 10 | 16 | 84 | 30 | 240 | 276 | 30 | 114 |

Step 3: Solving Multiple linear Regression using B1, B2 and a





Hence a0=6,a1=6,a2=0

Substituting above values in equation1 [Y=a0+a1x1+a2x2] we get

y=6+6(x1)+0(x2)

y=6(x1)+6 is the equation of multi linear regression.

USE CASES Of Multiple Linear Regressions:

Multiple linear Regression analysis is also used to understand which among the independent variables are related to the different parameter of dependent variable.

* Finding the missing values within the given data points
* Finding the loss value using slope line

For Example

Finding housing range of different parameter

Finding chances of a person’s tobacco consumption has any association with heart disease

Finding average point score per game in basketball

Finding crime rate in cities

Finding death rate in cities

To determine if a person is worthy for loan or not.

The immunity of a child can depend on the immunity of Parents

The height of a child can depend on the height of the mother, the height of the father, nutrition, and environmental factors.

The selling price of a house can depend on the desirability of the location, the number of bedrooms, the number of bathrooms, the year the house was built, the square footage of the lot and a number of other factors.

PYTHON CODE WITHOUT LIBRARY

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv("multivar.csv",sep='\t')

print(df.head(6))

#

X=df['Mech Apt'].values

X1=df['Consc'].values

Y=df['Job Perf'].values

n=len(X)

X\_mean=np.mean(X)

X1\_mean=np.mean(X1)

Y\_mean=np.mean(Y)

# # X.reshape(n,-1)

# # Y.reshape(n,-1)

#

sumX=np.sum(X)

sumX1=np.sum(X1)

sumY=np.sum(Y)

#

mulXY=np.multiply(X,Y)

mulX1Y=np.multiply(X1,Y)

#

sum\_mul\_XY= np.sum(mulXY) # - ((sumX\*sumY)/n)

sum\_mul\_X1Y=np.sum(mulX1Y) #- ((sumX1\*sumY)/n)

#

Xsquare = np.multiply(X,X)

X1square = np.multiply(X1,X1)

Ysquare = np.multiply(Y,Y)

#

sum\_X\_square=np.sum(Xsquare)

sum\_X1\_square=np.sum(X1square)

mul\_XX1 = np.multiply(X,X1)

sum\_mul\_XX1=np.sum(mul\_XX1) #- ((sumX\*sumX1)/n)

b1 = (((sum\_X1\_square)\*(sum\_mul\_XY))-((sum\_mul\_XX1)\*(sum\_mul\_X1Y))) / (((sum\_X\_square)\*(sum\_X1\_square))-(sum\_mul\_XX1\*sum\_mul\_XX1))

b2 = (((sum\_X\_square)\*(sum\_mul\_X1Y))-((sum\_mul\_XX1)\*(sum\_mul\_XY))) / (((sum\_X\_square)\*(sum\_X1\_square)) - (sum\_mul\_XX1\*sum\_mul\_XX1))

a=Y\_mean-(b1\*X\_mean)-(b2\*X1\_mean)

print("sum of Y =",Y)

print("sum of X =",X)

print("x mean=",X\_mean)

print("x1 mean=",X1\_mean)

print("y mean=",Y\_mean)

print("sum of X square =",sum\_X\_square)

print("sum of X1 square =",sum\_X1\_square)

print("sum of mul of X1\*Y =",sum\_mul\_X1Y)

print("sum of mul of X\*Y =",sum\_mul\_XY)

print("sum of mul of X\*X1 =",sum\_mul\_XX1)

print("sum of mul of X\*X1 square=",sum\_mul\_XX1\*sum\_mul\_XX1)

print(b1)

print(b2)

print(a)

updated\_y = b1\*X + b2\*X1 + a

PYTHON CODE WITH LIBRARY

# Encoding categorical data

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder = LabelEncoder()

X[:, 3] = labelencoder.fit\_transform(X[:, 3])

onehotencoder = OneHotEncoder(categorical\_features = [3])

X = onehotencoder.fit\_transform(X).toarray()

# Avoiding the Dummy Variable Trap

X = X[:, 1:]

# Splitting the dataset into the Training set and Test set

from sklearn.cross\_validation import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Fitting Multiple Linear Regression to the Training set

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X\_train, y\_train)

# Predicting the Test set results

y\_pred = regressor.predict(X\_test)

Visualization

