~~# Multiple Regression~~ Date - Feb 13

#Multiple linear Regression | model building

One of main importance of ML is that to find best accordance to someone on top of the dataset. So in multiple feature so some feature will get more weight in contrast to another on basis of specific needs.

At such case we are going to use multiple linear regression Eqn

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 \cdots\cdots + b_nx_n$$

(b_2: m_{22}, I.V.2)

- $\hat{y}$: dependent variable
- b_0 : m_1, I.V.1
- b_1x_1 : slope, Independent variable
- b_nx_n : n, n

I.V. ⇒ Independent variable

So In multiple linear regression each independent variable that is feature has own weight (that is. slope)

⊛ Assumptions of linear Regression

1) linearity
there should be linear Relation bet^n y and x

2) Homoscedasticity
there should be equal variance in dataset

3) Multivariate normality
There should be some normal relationship in distribution

4) Independentie
There should be no Pattern to be inside the data

5) Lack of multicollinearity ✓
Predictors should be correlated X1 & X2

6) The outlier check (not assumption)

a single outlier can make significance change in line from other points (It is Imp to check for outliers).

dummy variables

So as we know dataset has categorical variable in it so inorder to deal with it we have to use dummy variable

⇒ for numerical data we can directly assing some independent variable but for categorical we can't.

Ex:

State
Nyc
California
Nyc
California
California

⇒ using dummy variable

NYC	California
1	0
0	1
1	0
0	1
0	1

Dummy variable works like: switches

In above manner we can use ~~dum~~ categorical variable and as they are in numerical data we can able assign some independent variable to it.

→ for this chart we can use

b(n) D(n) → Inde(n)

So if Dn = 0 ⇒ Nyc Dn = 1 California so it is not case that we can have to use multiple Independent variable for a some set of value

⇒ There exist a case called dummy variable trap which we will learn further

Dummy variable Trap

Imp

we learn that we never include all dummy variables in eqn why? (There exist ~~to~~ which able to handle this

while working if you have n dummy variable only include (n-1) dummy variable in eqn

So If you have 2 set (different) so apply n-1 dummy variable rules to both of the sets.

Imp Always ommit one dummy variable.

P-Values

⊙ Statistical signifance

Say Toss a coin

p-value probability of a event happening

Hypothesis =>

H_0: This is a fair coin

H_1 = This is not a fair coin

=> we have to find which hypothesis true
=> after that consider a hypothesis among two & try to prove it wrong
=> so if it is wrong another is true use original is true

α = threshold of p-value & after it we will reject the hypothesis

Ex

	Heads ($)	Tails (₹) Pvalue
Toss 1	₹	0.5
Toss 2	₹	0.25
Toss 3	₹	0.125
Toss 4	₹	0.06
Toss 5	₹	0.03
Toss 6	₹	0.015

Pvalue decreasing

assume we are in H_0 enviroment

Say $\alpha = 0.05$ and after it we will reject hypoth

$\alpha = 0.5$

So as h_0 is false accordance to α value so H_1 is true.

~~that~~
So as $\alpha = 0.05$ so to $1 - \alpha$ = confident %

=> so rejecting hypothesis 1 with 95% Confidence

~~level~~ # we can set confidence level on our own

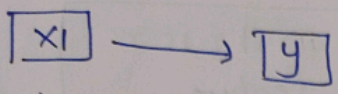
So it depends on the experiment

$\exists$ (α) is value which ~~start~~ make us suspicious about the hypothesis

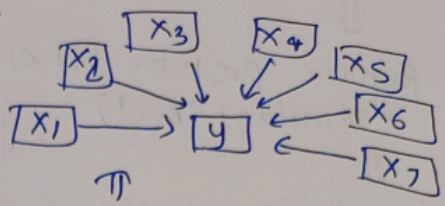
Multiple linear Regression Intitution

Here we will learn to build a model step to step.

for simple linear regression



single independent variable and on basis of it predict dependent variable



this is multiple linear Regression case

=) so now we have to select few among all and ignore all on basis of our need.

Reason can be

(A) we have to reduce garbage from model (unneccesary features) which make model unrealiable

(B) keep only important among a large set

#5 methods for building models (selecting Imp feature among all)

1] All in (throw all features in model building

2] Backward Elimination
3] Forward Elimination
4] Bidirectional Comparison
5] Score Comparison

⇒ Stepwise Regression

1] All in : you have a good data where all things are Important or prior knowledge or compulsion to use all

or preparing for backward elimination

2] Backward elimination: to stay model

step1 ⊙ select a significance level to fit in model (Stay in)

⇓

step2 ⊙ Fit the full model with all predictors

Step 3 : Consider the predictor with height p-value,
If p > SL go to step 4 else to Finish step → Finish

Step 4 : Remove predictor having value less than SL
(significant level)

{ . All we do in it is removal of feature

Step 5 : Fit the model without this variable
that is which go out will leave model &
we are going to rebuild the model.

and after rebuilding go to step 3 or, until.

So do it until our heighest p value will be less than significance level

(Just we do inside the toss problem as when . p<α we come out and that will be final model).

Forward selection

Step 1 :) Select significance level (SL) to enter the model → create simple regr for each feature with y

Step 2 ∋ Fill all simple regression models y ~ Xn, select one will lowest p value

Step 3 ∋ Keep this variable and fit all possible models with one extra predictor added, in one(s)

Step 4 :) Consider predictor with lowest pvalue If p<SL go to Step 3 else go to FIN

In forward all we do is adding the features

#Bidirectional model

Step 1 ⇒ Select Significant level to stay ~~and~~ and significant level to enter in model

SLEnter = 0.05 & SLSTAY = 0.05

↑ Forward selection — Backward selection

⇓

Step 2 : perform the next step of Forward selection (new variables must have p ~~<~~ SLENTER to enter)

⇓ add new variable on basis forward ~~of~~ Selection

Step 3 : Perform all backward Elm method (old variables must ~~sto~~ have p < SLSTAY to stay)

⇓

Step 4 : No new variables can enter or leave so we can exit. ⇓ FIN: model is ready

⇒ with forward selection adding variables on basis of SLENTER and removing variable on basis of SLSTAY (i.e backward selection)

#All possible

most resource consuming approach

step 1 : select criteria of goodness of fit

⇓

step 2 : Construct all possible Regression model : $2^n - 1$ Combinations

⇓

step 3 : select one with best criteria

⇓

FIN model is ready

Tip:
So If you have 10 Columns you have 1023 models

⇒ this is resource consuming

⇒ we are going to use backward elimination as it is fastest one to build a model.

Coding Part multiple linear Regression

We want to findout best startup to invest on basis of dataset

Step A ⇒ Preprocessing

⊙ Encoding
in one hot encoder & column transformer we pass X as a whole

⊙ Feature ~~selection~~ scaling (Data | Performance will give info about model)

In multiple linear Regress we don't have to apply for feature scaling

If your data is not linear so it will perform poorly for multiple linear Regression model

⇒ you don't have to check for assumptions of MLR for dataset, as if assumptions are not true the model will perform badly

⇒ It will be waste of time to check for MLR assumptions

Q) do we have to worry about dummy variable trap?

⇒ no model take care about it

do we have to apply backward elimination?

no sklearn will take of everything for you.

Code for model building for MLRM

class for MLRM is same ~~for~~ that of (SLR)

So code

```
from sklearn.linearModel import LinearRegress
Lr = linearRegression()
Lr.fit(xtrain, ytrain)   & that will build your model
```

Prediching Test results

plot real profit vs predicted

ypred = regressor. predict (xtest

~# print throught np precision upto 2points

np. set_printoptions(precision = 2)

now concatnation of two vectors

print (np.concancate ((y predicted. reshape(len(y-pred)),1))

↑ Reshape ypred

↑ tupple

insert() Imp 4 ea

⇒ for sidewise view

m = pd.dataframe()

m. insert

take 3 argument
column = "name of column")
value = np array
loc ⇒ column no

⊛ To ~~set~~ Round up values in np array you can use
set_printoption (precision = 2)

you can also use round function for this