M T W T F S S
Page No.:
Date: 22/12/21
YOUVA

Machine Learning - 1

Crisp-DM Framework

- Business Understanding
- Data Understanding
- Data Preparation
- Model Building
- Model Evaluation
- Model Deployment

EDA & Statistics (Previous Course)
- → Visualisation Data
- → Cleaning Data
- → Performing EDA

Predictive Analytics → Use cleaned and prepared data & build models
- Linear Regression → One of most widely used models → Used for linear relationship between variables
- Logistics Regression → used to predict class or category → Most widely used classification model
- Clustering → Exploring in absence of labelled data
- PCA → Perform dimensionality reduction

Pre-requisites

1. Exponentials
2. Logarithms
3. Equation of a straight line
4. Basis Transformation
5. Eigen decomposition
6. Inferential Statistics
7. Hypothesis Testing

Types of Machine Learning Algorithms

1. Regression → O/P variable to be predicted is a Continuous Variable
2. Classification → Predict a Categorical variable i.e Ex: Spam or Ham
3. Clustering → No notion of a label ... allocated to groups/ cluster formed. Ex: Customer Segmentation

Classification	Clustering
→ Labels (Spam & Ham example)	→ No Labels Ex: Type of cluster

Machine Learning - 1

Crisp-DM Framework

- Business Understanding
- Data Understanding
- Data Preparation
- Model Building
- Model Evaluation
- Model Deployment

EDA & Statistics (Pradhna Course)

- Visualisation Data
- Cleaning Data
- Performing EDA

Predictive Analytics → Use Cleaned and prepared data & build models

- Linear Regression → One of most widely used models → Used for linear relationship between variables
- Logistic Regression → used to predict class or category → Most widely used classification model
- Clustering → Exploring in absence of labelled data
- PCA → Perform dimensionality reduction

Pre-requisites

1. Exponentials
2. Logarithms
3. Equation of a straight line
4. Basis Transformation
5. Eigen decomposition
6. Inferential Statistics
7. Hypothesis Testing

Types of Machine Learning Algorithms

1. Regression → O/P variable to be predicted is a Continuous Variable
2. Classification → Predict a Categorical variable. e.g. Ex: Spam or Ham
3. Clustering → No notion of a label ... allocated to groups/ Clusters formed. Ex: Customer Segmentation.

Classification	Clustering
→ Labels (Spam & Ham example)	→ No Labels Ex: Type of cluster

Machine Learning - 1

Crisp-DM Framework

- Business Understanding
- Data Understanding
- Data Preparation
- Model Building
- Model Evaluation
- Model Deployment

EDA & Statistics (Prerequisite Course)

- Visualisation Data
- Cleaning Data
- Performing EDA

Predictive Analytics → Use Cleaned and prepared data & build models

- Linear Regression → One of most widely used models → Used for linear relationship between variables
- Logistics Regression → Used to predict class or category → Most widely used classification model
- Clustering → Exploring in absence of labelled data
- PCA → Perform dimensionality reduction

Pre-requisites

1. Exponentials
2. Logarithms
3. Equation of a straight line
4. Basis Transformation
5. Eigen decomposition
6. Inferential Statistics
7. Hypothesis Testing

Types of Machine Learning Algorithms

1. Regression → O/P variable to be predicted is a Continuous variable
2. Classification → Predict a Categorical variable. Ex: Spam or Ham
3. Clustering → No notion of a label, allocated to groups/clusters formed. Ex: Customer Segmentation

Classification → Labels (Spam & Ham example)

Clustering → No Labels Ex: Type of cluster

Supervised and Unsupervised Learning

		Labels	Types of Labels
Supervised Learning Method	Regression	✓ Predicting scored students	Continuous
	Classification ✓	Spam or ham from past email data	Categorical

↳ Past data with labels is used for building the model

Unsupervised Learning Method — Clustering

↳ No pre-defined labels are assigned to past data

Regression Algorithms

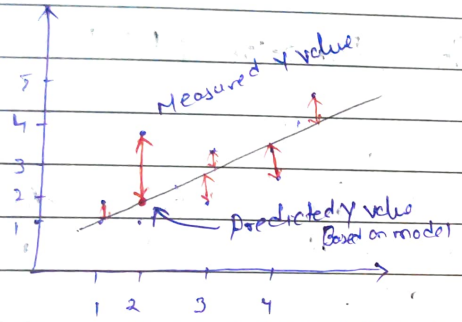
→ It has labeled data

Types of Regression

1. Simple Linear Regression
   ↳ Only 1 independent variable
2. Multiple Linear Regression
   ↳ Model with more than 1 independent variable.

1. Simple

Best - Fit Line



→ At $x=2$, measured y value = 4
, Predicted y value = 2

→ Difference betn measured and predicted value is called Residuals
$E_x = 4-2 = 2$ (residual of value)

→ Syntax ⇒ $\boxed{e_i = y_i - y_{pred}}$ It is also called as Error

→ Ordinary Least Squares Method
It is a method of minimising the total error square.
$e_1^2 + e_2^2 + e_3^2 + \cdots + e_n^2$ (RSS Residual Sum of Squares)

Coefficient of least Squares regression line are calculated by Ordinary Least Squares method.

M T W T F S S | Page No.: | Date: | YOUVA

→ We need to find the best value of RSS which will best fit to the eqn of straight line.

$y = \beta_0 + \beta_1 x$

Intercept ↓ ↓ → Slope

~~eqn~~ ~~$\beta_0 + \beta_1 x_i = y_i - y_{pred}$~~

$e_i = y_i - y_{pred}$

$e_i = y_i - (\beta_0 + \beta_1 x_i)$

$e_i = y_i - \beta_0 - \beta_1 x_i$

$RSS = (y_1 - \beta_0 - \beta_1 x_1)^2 + (y_2 - \beta_0 - \beta_1 x_2)^2 + \cdots + (y_n - \beta_0 - \beta_1 x_n)^2$

$$RSS = \sum_{i=1} (y_i - \beta_0 - \beta_1 x_i)^2$$

→ We need to find the best possible of RSS and we know y_i and x_i but we need to find the optimal value of β_0 and β_1.

→ It is an absolute quantity, so changing to unit of dependent variable, RSS Changes.

* Ways to minimize Cost Function (RSS)

1. Differentiation
2. Gradient Descent Function
   → It is an optimisation algorithm which optimizes the objective funtn (for linear regression it's a Cost Function) to reach to the optimal soln.

TSS

→ RSS is an absolute quantity, so Changing to unit of dependent variable the RSS Changes

→ Hence we need a relative approach instead of RSS (absolute approach).

→ 1. Calculate the $\bar{y}$ (y avg) of all the given points.

2. $$TSS = \sum_{i=1} (y_i - y_{avg})^2$$

Intercept value = $\bar{y}$

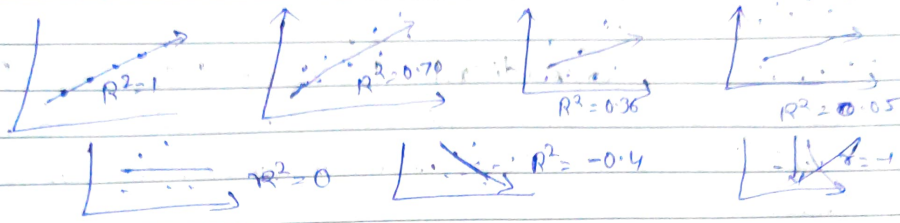
R^2 = Tells how good our model is.

$$R^2 = 1 - \frac{RSS}{TSS}$$

$\frac{RSS}{TSS}$ = normalised Quantity which tells how well our model is fitted.

High value of R^2 is good.

→ In correlation coefficient and R^2 square value is same.



RSE (Residual Sum of Square)

→ $RSE = \sqrt{\frac{RSS}{df}}$, $df = n-2$ (n = no. of data points, df = degree of freedom)

→ It also has same disadvantage of RSS.

→ R value lies beto 0 – 1.

1 – variance in the data being explained by the model

0 – None of the variance value is being explained by the model

→ R^2 = (Correlation) of a straight line

→ Slope doesn't determine the correlation between variable

→ Correlation Coefficient lies beto (−1, 1)

→ R^2 = How much variability in y we are able to explain with the model.

$R\ Squared = \frac{MSS}{TSS} = 1 - \frac{RSS}{TSS}$ (Model Sum of Squared / Total Sum of Squared)

(variability in y explained by model / (total variability in y)) X Not understood

MSS + RSS = TSS

M T W T F S S
Page No.
Date 24/12/21
YOUVA

Gradient Descent (Steepest Descent)

→ It is an iterative 1st-order optimisation algorithm used to find local minimum/maximum of a given function.

→ It is a method to minimise a cost/loss function.

→ Cost funto quantifies the error betn predicted value and expected values and present it in the form of a single real no.

→ It was originally proposed by CAUCHY in 1847.

Function Requirements

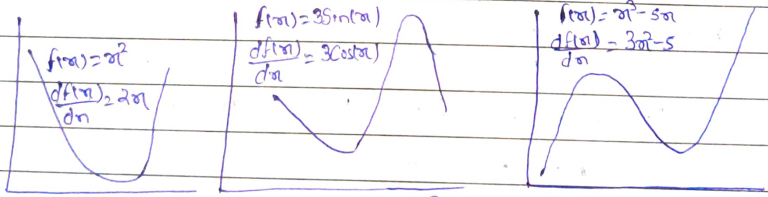
→ It does not work for all functn.

→ There are two specific requirement

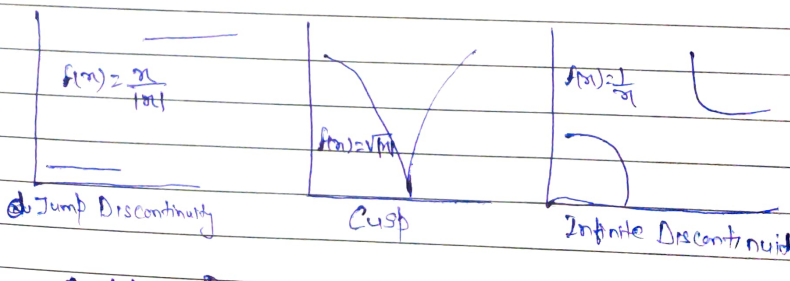
1. differentiable
2. Convex

1. Differentiable

→ If a funto is differentiable it has a derivatives for each point in its domain. not all funto meet these criteria.



↑ Differentiable Functions ↑



Jump Discontinuity — Cusp — Infinite Discontinuity

↑ Non-Differentiable Functn ↑

M T W T F S S

Page No. YOUVA

Date 24/12/21

Gradient Descent (Steepest Descent)

→ It is an iterative 1st-order optimisation algorithm used to find local minimum/ maximum of a given function.

→ It is a method to minimise a cost/loss function.

→ Cost funto quantifies the error betn predicted value and expected value and present it in the form of a single real no.

→ It was originally proposed by CAUCHY in 1847

Function Requirements

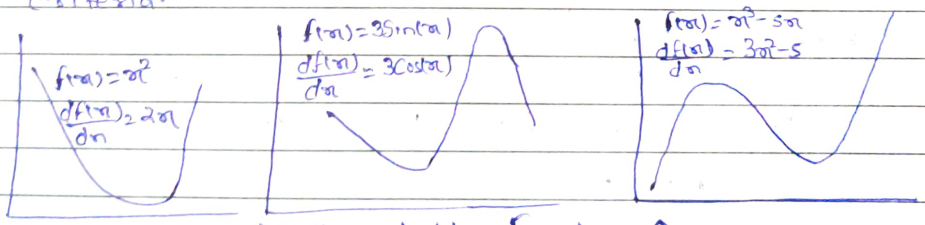
→ It does not work for all functo.

→ There are two specific requirement

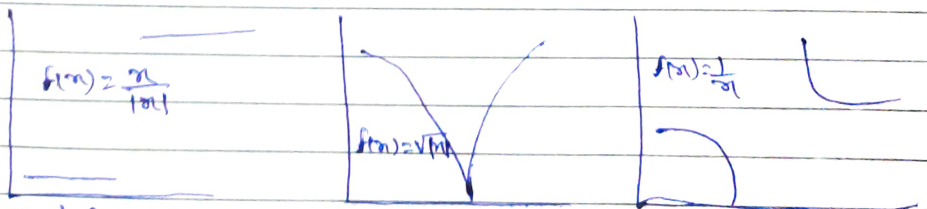
1. differentiable
2. Convex

1. Differentiable

→ If a funto is differentiable it has a derivatives for each point in its domain. not all funto meet these criteria.



↑ Differentiable Functions ↑



Jump Discontinuity — Cusp — Infinite Discontinuity

↑ Non- Differentiable Functo ↑

2. Convex

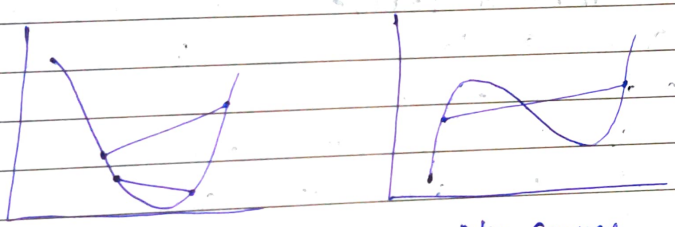
① → For a univariate funto, it means that the line segment connecting 2 function's points lays on or above its curve (it doesn't cross it).

→ If it does it means that it has a local minimum which is not a global one.

→ 2 point x_1, x_2 laying on the function's curve this condition is expressed as.

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

λ = Denotes a point's location on a section line and its value has to be beto 0 (left point) and 1 (right point)



Convex Non-Convex

2. If a univariate funto is convex is to calculate the second derivatives and check if its value is always bigger than 0.

$$\frac{d^2 f(x)}{dx^2} > 0 \quad \text{(strictly convex)}$$

→ Quasi-convex funto can be used with Gradient descent algorithm and they are so called ~~cradle~~ saddle point (called minimax points) where algorithms can get stuck.

1st Derivatives

$$f(x) = x^4 - 2x^3 + 2$$

$$\frac{df(x)}{dx} = 4x^3 - 6x^2 = x^2(4x-6) \quad \begin{cases} 0 & x=0 \\ 0 & x=1.5 \end{cases}$$

Note:- → At $x=0$, $x=1.5$ the first derivatives is 0 so this places are candidates for funtn's extrema (min^m or max^m)
→ Slope is 0 there

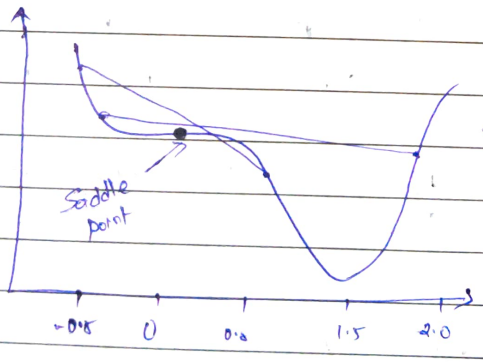
2nd Derivatives

$$\frac{d^2f(x)}{dx^2} = 12x^2 - 12x = 12x(x-1)$$

$$\frac{d^2f(x)}{dx^2} = 12x(x-1) \quad \begin{cases} 0, & x=0 \text{ and } x=1 \end{cases}$$

→ The local where Second derivative is 0 are called an inflexion point, a place where the curvature Changes sign (means it Changes from convex to concave and or vice-versa.

* for $x<0$: funtn is convex
* for $0<x<1$: funtn is concave
* for $x>1$: funtn is convex again



At $x=0$, 1st & 2nd derivative is 0 so it's called as saddle point
$x=1.5$ is global minimum.

M T W T F S S
Page No.
Date
YOUVA

Gradient

→ It is a slope of a curve at a given point in a specified direction.

→ In case of a univariate funto, it is simply the 1st derivatives at a selected point

→ In case of a multivariate funto it is a vector of derivatives in each main direction (along variable axis)

→ A gradient for an n-dimensional funto f(x)

$$\nabla f(p) = \begin{bmatrix} \frac{df(p)}{df_1} \\ \vdots \\ \frac{df(p)}{df_n} \end{bmatrix}$$

Example:- $f(x) = 0.5x^2 + y^2$

$$\frac{df(x,y)}{dx} = x \quad ; \quad \frac{df(x,y)}{dy} = 2y$$

$$\nabla f(x,y) = \begin{bmatrix} x \\ 2y \end{bmatrix} \qquad [10,10] \Rightarrow \begin{bmatrix} 10 \\ 20 \end{bmatrix}$$

→ Slope is twice steeper along the y axis.

Gradient Descent Algorithm

→ It iteratively calculates the next point using gradient at the current position, then scale it (by a learning rate) and substracts obtained value from current position.

→ It substracts the value because we want to minimise the funto (to maximise it would be added)

$$p_{n+1} = p_n - \eta \nabla f(p_n)$$

η = Scales the gradient and control step size (learning rate and have strong influence on performance)

Advanced Gradient-based optimisation Method
↳ Momentum or Nesterov (Accelerated) Gradient Descent
↳ RMSprop, ADAM or second-order ones like Newton-Raphson

Page No.:
Date:
M T W T F S S
Algorithm
YOUVA

Note:-
→ If the learning rate is too high, we might OVERSHOOT the minima and keep bouncing, without reaching the minima.
→ If learning rate is too small, the training might turn out to be too long or may reach maximum iteration before reaching the optimum point.

In Summary, Gradient Descent Method's steps are:
1. Choose a starting point (initialisation).
2. Calculate gradient at this point.
3. Make a scaled step in the opposite direction to the gradient (objective: minimise)
4. Repeat points 2 and 3 until one of the criteria is m
   * max no. of iterations reached
   * step size is smaller than the tolerance.

M T W T F S S
Page No.: YOUVA
Date: 24/12/21

Simple Linear Regression in Python

Making Predictions

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$ (^ mean estimate i.e. estimated value of y, β_0, β_1)

$$e = y_i - y_{pred}$$

$$RSS = \sum_{i=1}^{n} (y_i - y_{pred})^2 = \sum_{i=1} (y_i - \beta_0 - \beta_1 x_i)^2$$

* Model's Formulation - $Y = \beta_0 + \beta_1 X + \epsilon$
* Fitting a line :- minimize the sum of squared errors (residuals)
* Assessing goodness of fit :

  R squared = (MSS/TSS) = 1 - (RSS/TSS)

* Making a predictions

  For a new instance, predict the output as $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

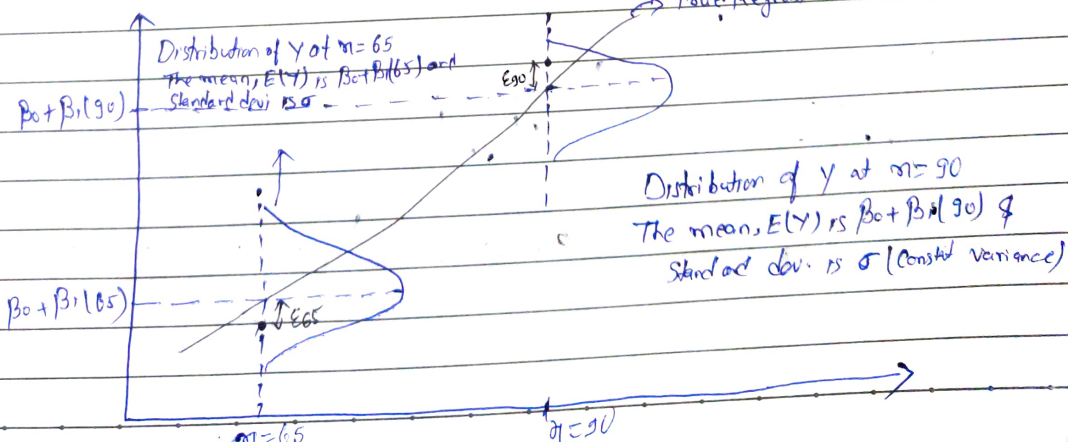
* $t = \beta_i / SE(\beta_i)$ => t-value for a particular coefficient is given by its coefficient divided by its standard error

Assumptions of Simple Linear Regression

-> We try to find the data of population by using sample data to make a model.

-> Due to which we add some uncertainity in the output from the model.

* In linear regression, at each X, finds the best estimate for Y.
  - At each X, there is a distribution on the value of Y.
  - Model predicts a single value, therefore there is a distribution of error terms.



Note:- There is no assumption on the distribution of x and y.

M T W T F S S
Page No.:
Date:
YOUVA

→ We know for every value of X we can have different value of Y, but the model gives (Predict) one value of Y when there are many and this introduces some error.

* Assumption

1. → There is a linear relationship between X and Y.

2. → Error terms are normally distributed (with mean zero) (not X & Y)

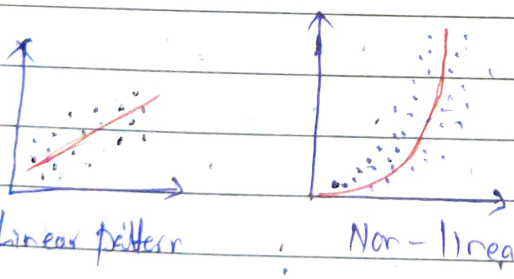
- Mean for one value of X we can can get different value of Y. We chose one value of Y, so there is an error for each chosen value of Y that follows a normal distribution.
- For every x there is a distribution of error.

3. → Normal distribution of error, the std. deviation is same across different value of X. i.e std. dev σ is same at x=65 or 90.
(Error has constant deviation or variance across different value of X.)

4 → Error term is independent of each other
- Mean previous doesn't have any effect on present error.

5 → Error terms have constant variance (homoscedasticity)

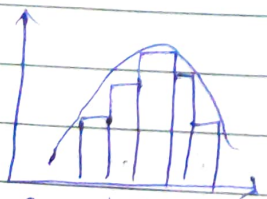
①



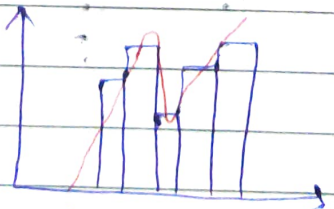
Linear pattern

Non-linear pattern

②



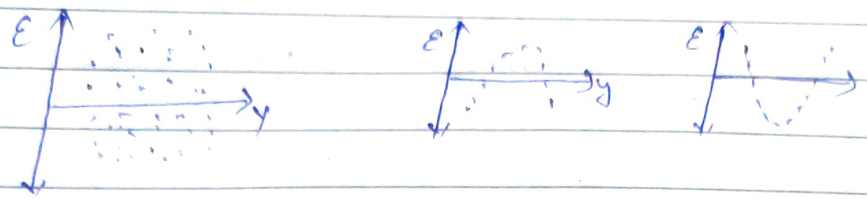
Error term normally distributed



Error term not normally distributed.

→ Not understood

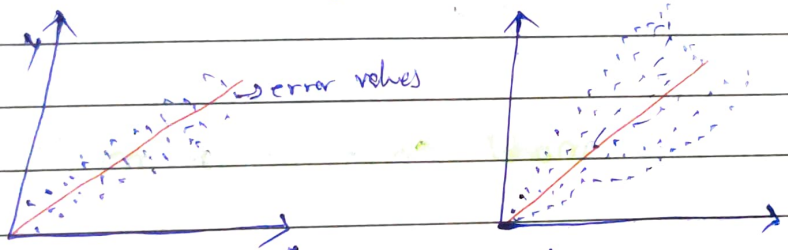
3



No visible pattern
- Error terms independant

visible Pattern - Error terms dependent

5. → The variance should not increase (or decrease) as the error value changes
→ The variance should not follow any pattern as the error terms change.



Constant variance (Homoscedastic)

Changing variance (Heteroscedastic)
↳ Inferences made on the model would be unreliable
↳ Even if we fit a line through the model, we can't make inference about model.
↳ Parameters used to make inferences will become highly unreliable.

Steps:
1. Reading and understanding the data
2. Training the model
3. Residual Analysis
4. Prediction and evaluation on the test set

M T W T F S S
Page No.:
Date: 26/12/21
YOUVA

Building a Linear Model

TV Radio Newspaper Sale

Predictable variable. → Target Variable

$$y = c + m_1x_1 + m_2x_2 + \dots + m_nx_n$$

y = is the response
c = intercept
m_1 = Coefficient of 1st feature
m_n = Co-efficient of nth feature

"m values are called model coefficient or model parameters"

Note:- Linear regression model can be built with "statsmodel" and "sklearn" packages module/library

Steps to Create a Model

1. Create predictable variable(x) and target variable (y)
2. Create train and test (unseen or hidden dataset) test set (70-30 or 80-20)
3. Train model on training set (i.e learn the coefficients)
4. Evaluate the model (training set, test set)

1. X = df["xyz"] , y = df['abc']
2. import Sklearn,

```
from Sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.70, random_state = 100)
```

train_size of train set is 70% (and test set is 30%)

random_state of test set (any thing will)

$p = .001$ (very low)

M T W T F S S
Page No.:
Date:
YOUVA

3. Train model

3.1. "Statsmodel" by default exclude "c" (intercept) while training the model.

3.2. We need to explicitely add "c" value to the model.

```
from statsmodel.api as sm
x_train_sm = sm.add_constant("x_train")
```

It will add a column "const" to the dataframe of "x_train" with Const value = 1.

Const	x_train
1.0	2
1.0	5
⋮	⋮
1.0	10

If Const is not added then the line will pass through origin. (by default)

$y = c.const + m1.x_1$

3.3. Fitting the model (OLS = (ordinary least Square))

- `lr = sm.OLS(y_train, X_train_sm)`
  - ↳ It is a linear regression object.
  - ↳ No learning of coeffcia is done overhere only object (or instance) of Class OLS is created.
- ~~`lr_model.params`~~
- `lr_model = lr.fit()` // learny of coefficial
  `lr_model.params` // returns c and m (intercept & slope)

3.4. `lr_model.summary()`

- Statsmodel gives detailed Summary which is not provided with sklearn (Sykit learn)

→ R-squared = 0.816 ie 81% of variance in the target variable is due to computational variable.

→ Coef and p-value

→ R-Squared is 81.6% very high

→ (F-Statistics) Prob ↓ mean the fit is not purely by chance.

- If Prob(F-Statistics) < 0.05 then overall model fit is significant

→ R squared varies from 0 to 1
R-squared 0 means none of the variance in the data is explained
1 means all of the variance in the data is explained.

4. Residual Analysis and Predictions & Evaluation

Residuals
⇒ Now we need to verify our assumption that the (error) errors are normally distributed with mean equal to 0.
⇒ Errors should be independent of each other.
⇒ Look for patterns in residuals (we should not be able to identify)

```
y_train_pred = lr_model.predict(X_train_sm)
```

```
res = y_train - y_train_pred
sns.distplot(res) # normal distribution of "res"
plt.scatter(X_train, res)
```

Predictions
⇒ Predict the y value of test set
`X_test_sm = sm.add_constant(X_test)` → add const value to test set to make it compatible for model prediction
`y_test_pred = lr_model.predict(X_test_sm)`

Evaluation
Test set `r2_score(y_true = y_test, y_pred = y_test_pred)`
Train set `r2_score(y_true = y_train, y_pred = y_train_pred)`
⇒ r2-Score value tells us how well the model is performing on test set.
⇒ If r2-Score of test set value lies with 10% difference of r2-Score of train-set then we can say that the model is very nicely able to predict the output on the test set.

Note:- ① p-value need to be very small (less than 0.05 / 5%), then we can reject the Null Hypothesis and conclude that the coefficient is significant.
② Sum of residual should be equal to zero, as residuals are normally distributed around zero (mean = 0)

(P.T.O.)

Note:- ① Low value of RSME (Root Mean Square Error) is better as it tells the deviation of the predicted value of a model from the actual value.

Linear Regression using Sklearn

⇒ Sklearn is a goto package for predictive analysis and machine learning.

Steps

1. Create an object of linear regression
2. Fit the model
3. See the params, make predictions (train, test)
4. Evaluate (r2, etc)

1.

X_train_lm = X_train_lm.values.reshape(-1,1) → ignore all the rows, insert 1 col

y_train need to be single series

Note:- ③ R-Squared value gives the extent of the fit. i.e. how much variance in the data is being explained explained by the model. It doesn't tell anything about the significance of fit.

④ F-Statistics

→ It determines the overall model fit better or not.

→ The idea behind F-test is that it is relative comparison betn the model that we have built and the model without any of the coefficient except B_0.

↑ F-Stats then Prob(F-Stats) ↓ → Model is significant

↓ F-Stats then Prob(F-Stats) ↑ (0.05) → Model fit is insignificant and the intercept only model can provide a better fit.

M T W T F S S

Page No.: YOUVA

Date:

5. t-value of the coefficient.

6. Popular Package to built linear Regression Model.
   1. Statsmodel-api
   2. SkLearn
   3. SciPy

7. Correlation coefficient value shows how strongly or loosely the relationship between 2 variables are related to

8. Null hypothesis states that the coefficient B_1 is equal to zero. (no statistically significant relationship)
   Alternate hypothesis is $B_1 \neq 0$

9. Correlation coefficient should be in range [1, -1]
   value beyond this range indicates an error in measurement

10. R-Squared why it is called that. (on Simple Linear regression)
    (Correlation) → Corrs = np.corrcoef(X-train, y-train) → correlation coefficient
    ∴ R-Squared = Corrs[0,1] **2
    Correlation (Pearson) is also called "r" or "Pearson's R"

11. There is no notion of standard or good RSME value.
    → RMSE depends on the units of the Y variables
    → RMSE is not normalised measure
    For 2 different model with same dataset to model with lower RMSE is better

12. Scaling does not affect the model statics and the model goodness will remain the same.
    → Scaling helps with interpretation interpretation becomes easy when we have different feature all on a similar scale.
    Mean = 0 or Stdev = 1
    → Scaling uses gradient descent at background so gradient descent will be faster when variable we have, have similar scale.
    Note: when we scale X, Y then the slope or Beta coefficient will be same to correlation beta X & Y.

M T W T F S S
Page No.
Date 28/12/12
YOUVA

Multiple Linear Regression

→ It gives the relationship between two or more independent input variables and a response variable.
→ It is used when one variable is not sufficient to create a model and make accurate predictions.
→ Adding the R squared value of 2 or more variable doesn't decreases, either it will increase or remain same.

* Formulation of MLR

Extensions of Simple Linear Regression to 'adds' more factors

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

* Interpretation of the Coefficient

→ Change in Y (E(Y)) after per unit increase in the variable (X_s) when other predictors are held constant.

* A lot of ideas are same as or are simple extensions of SLR.
→ Model now fits a "hyperplane" instead of a line
→ Coefficient still obtained by minimizing sum of Squared error (Least Squares Criterion)
→ For inference, the assumption from SLR still hold
- Zero mean, independent, Errors are Normally distributed that have Constant variance.

Moving from SLR to MLR: New Considerations

* New Considerations:
1. Adding more isn't always helpful
   a. Model may "overfit" be becoming complex
      (i) Model fits the train set "too well" (memorizing the data) and doesn't perform well on test set or other set.
      (ii) Symptoms: high train accuracy, low test accuracy
   b. Multicollinearity
      1. Associations between predictor variables

2. Feature selection becomes an important aspect, so that 'overfit' should not happen and multiple variable increases the accuracy of the model is a generic way.

Multicollinearity

→ It refers to the phenomenon of having related predictor variables in the input dataset.

→ In simple terms, in a model which have been built using several independent variables, some of those variables might be interrelated, due to which the presence of that variable in the model is redundant.

→ We can drop some of these related independent variables as a way of dealing with multicollinearity.

Effect

* Interpretation:-
  → Does "Change in Y, when all the other are held Constant" apply? No as the other variable are dependent.
* Inference
  → Coefficients swing wildly, signs can invert.
  → p-value are, therefore, not reliable.

Does not Effect:
→ The predictions, precision of the predictions
→ Goodness-of-fit statistics

Multicollinearity is a big issue when we are trying to interpret the Model but detailed is required

Two Basic way of dealing Multicollinearity

1. Pairwise Correlation
   → Looking at the correlation between different pairs of independent variables.
   → Scatter plots to visually inspect.
2. Correlations to quantify the linear association.
   (Variance Inflation Factor (VIF))

--> Sometimes pairwise correlation is not enough.
→ Instead of one variables, the independent variable might depend upon a combinations of other variables.
→ VIF Calculates how well one independent variable is explained by all the other independent variables combined

$$VIF_i = \frac{1}{1-R_i^2}$$

i = i^{th} variable which is being represented as a linear combination of rest of the independent variable.

* Common heuristic for VIF values
  - >10: Definitely high VIF value and the variable should be eliminated
  - >5: Can be okay, but is worth inspecting
  - <5: Good VIF value. No need to eliminate the variable.

Note:- VIF doesn't have to do anything with target variable.

Q

Yes, VIF (Will) can change after dropping an independent (Predictable) variable as we are building a model but by coincidence if we droped a variable which doesn't have any relation with the variable while VIF was being calculated but in general VIF will drop. so it a good practise to recheck your VIF again-again after droping a variable.

Ex:- Suppose variable = $x_1, x_2 \dots x_{10}$

VIF of x_1 calculated after droping x_7 to the VIF of x_1 will change (decrease) in general.

-> We should have minimum no. of variable to make to build a lean model
-> In case of interpretation we must deal with multicollinearity

M T W T F S S
Page No.:
Date:
YOUVA

Dealing With Multicollinearity

1. Dropping Variables
   -> Drop the variable that is highly correlated with others as other variables gives the same information.
   -> Pick the business interpretable variable (if interpretation and explicability important)

Ex:- Suppose 2 variable are strongly related with one and another, the we should drop a variable which is of lest important from business perspective.

2. Create a new variable using the interactions of the older variables. (same sample given above)
   -> Add interactions features, drop original features

Ex:- From 2 variable, make a new variable which Combine the information of both & multicollinear variable. (it has some implications)

-> Variable Transformations (get all & variable and make a new variable from it)
   * PCA (Principle Component Analysis)
   * PLS (Partial least Square).

Notes:- 1. Multicollinearity makes some variable redudent hence the p-values charges.
2. Predictive power given by R-Squared value is not affected because even though we migh have redundent variable in the model.

Dealing with Categorical Variables

Handling Categorical Variables With few levels

Value	(Dummy) Indicator Variable
Gender	Female
Male	(dummy value) 0
Female	(dummy value) 1

Value	Indicator Variable	
Furnishing status	Furnished	Un-furnished
Furnished	1	0
Semi-furnished	0	1
Unfurnished	0	0

$$x_i = \begin{cases} 1 & \text{if ith person is female} \\ 0 & \text{if ith person is male} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \begin{cases} \beta_0 + \beta_1 + \epsilon & \text{; if ith person is female} \\ \underbrace{\beta_0 + \epsilon}_{\text{(Base State)}} & \text{; if ith person is male} \end{cases}$$

→ When x is 0 with any no. of variable.

If the person is a female it has an extra affect (β_1) compared to base state.

Note:- Creation of dummy variables to convert a categorical variable into a numeric variable is an important step in data presentation.

Formula:- N no. of Categorical Variable

(N-1) no. of dummy variable will be created

* → In case of multiple variable with different levels then we need (n-1) dummy variable [n = level of Categorical Variable]

Feature Scaling

→ Why we scale feature?

1. Ease of interpretation

When we have multiple variable with different scale level then it becomes difficult to for co-efficient to signify importance of them. But if we get all the variable on one scale then it becomes easy to compare the coefficient of one with another hence Interpreted become easy.

2. Faster Convergence for gradient descent methods

When we bring (scale) all the variable in a perticular (nearby) range i.e (-1,1), (0,1) (-3,3) which is nearby then it convergence for gradient descent method becomes faster.

Note:- Neither p-values nor Model Accuracy changes with scaling, it just changes the co-efficient which is upto our interpretation.

Popular Feature Scaling Methods

1. Standardization ($x = (x - \text{mean}(x)) / \text{st.dev}(x)$)
2. MinMax Scaling ($x = (x - \min(x)) / (\max(x) - \min(x))$)
   (Normalization) (0-1) range (Mean = 0, Std dev = 1)

2. Scaling should be done after train-test split because we don't want test dataset to learn anything from the train data. If done before then test data will have information regarding the minimum & maximum values.

Read the extra topic given at the end of this section.

M T W T F S S
Page No.
Date 31/12/21
YOUVA

Model Assessment and Comparison

→ In multiple linear regression, we may build more than one model and compare them to check which one yields optimal result.

Q. Shall we compare the F statistics, R-Square of different model

Selecting the best Model

→ Trade off between explaining highest variance and keeping it simple. (Bias vs Variance Trade-off)

(Key idea: Penalize models for using higher no. of predictors)

→ We can have a perfect model with many variable but it may overfit or keep simple model with less required no. of variable and predict the model by keeping it simple

→ Now we need to make the trade off by penalizing the model with high no. of variable.

Suppose:- We have 2 model with same R-Squared one with high no. of variable other with less no. of variable, so in that case we reduce the R-Squared with less no. of variable (adjust the R-Square)

$$\text{Adjusted } R^2 = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$$

(↓ Record ↑ i/p variable is Not good)
(↑ record ↓ i/p variable with same R-Squared is a better option)

Akaiken Information Criteria (AIC)

$$AIC = n * \log\left(\frac{RSS}{n}\right) + 2p$$

(Low value of AIC is better hence "+ve" penality)
(Penality increase linearly with ↑ no. of i/p variable in the model)

→ It is used in automatic model selection

→ BIC, Mallows Cp and many more.

Note:- BIC is similar to AIC, it just put harder penalty on the predictor.

N, n = Sample Size
p = no. of predictor variables.

Read about AIC, BIC, Mallows' Cp from given link.

Note:- → R-squared doesn't penalise the model for having more no. of variables.

→ Adding variable to the model, the R-squared will always increase (remains constant if correlation beto that variable and dependent variable is zero).

→ R-squared assumes that that any varrable adde to the model increases the predictive power.

→ Adjusted - R-squared on the other hand, penalise the models based on the no. of variables present in it.

→ So if we add a variable and the Adjusted R-squared drops, we can be sure that the variable is insignificant to the model and shouldn't be used.

→ In case of multiple linear regression, we should always look at the adjusted R-squared value in order to keep redundant variables out from your regression model.

Feature Selection

→ In case of multiple linear regression we may have a few potential predictor variables. Selection those is an important task.

Try all possible combinations? 2^p models for p features

1. Manual Feature Elimination:
   - → Build Model
   - → Drop features that are least helpful in prediction (high-p-value)
   - → Drop features that are redundant (using Correlation, VIF)
   - → Rebuild model and repeat.

Note:- Manual feature Elimination is a good Choice when we have less feature suppose 10 or 20 but it is not practical approach once we have a large no. of features like. 50, 100 etc and so on.

2. Automated Approach
   - → Top 'n' feature: RFE (Recursive Feature Elimination)
   - → Forward/Backward/Stepwise selection: Base on AIC
   - → Regularization (Lasso)
     (It makes the Coefficient 0 for the variable which is not needed for the model)

Note:- 1. Backward and Stepwise Selection gives same output but later method is more popular.

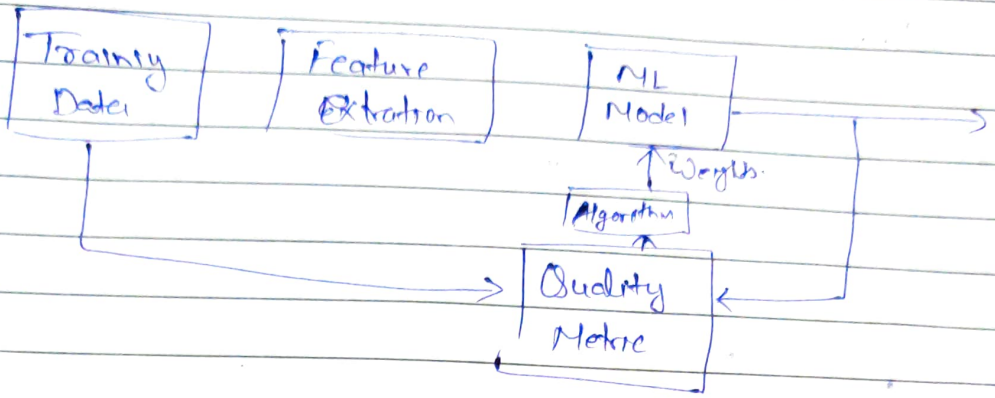
→ A balanced approach: use a Combination of automated (coarse tuning) elimination + manual (fine tuning) selection.

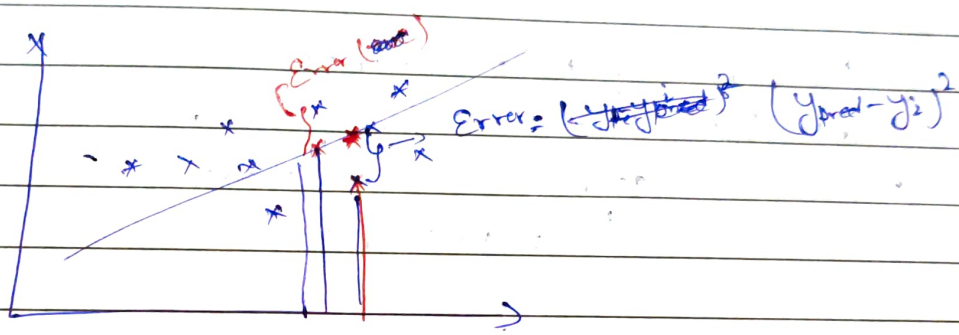
Note:- After automatic elimination of variable we need to use our expertise and subjectivity to eliminate a few other features (fine tuning).

Note:- → High p-value is indicative of the variable not having predictive power.

→ It might not necessarily mean that the variable is being described by one or more of the feature variable.

SGL-ML-C34-CC-G04



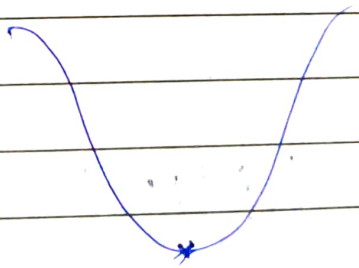


$$\text{All data} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

RSS → We amplify the error and our gradient descent can work properly on it.

Gradient Descent

- $f(x)$ →



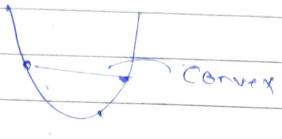
$f'(x) \Rightarrow \frac{d f(x)}{dx} = 0$? find x will give minimum

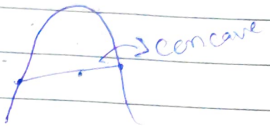
$f''(x)$ $\frac{d^2 f(x)}{dx^2}$

* For linear regression start point doesn't affect to ~~local~~ global minima

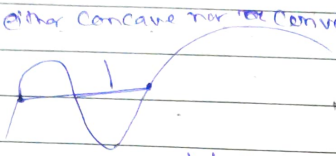
M T W T F S S
Page No.:
Date:
YOUVA

Convex & Concave

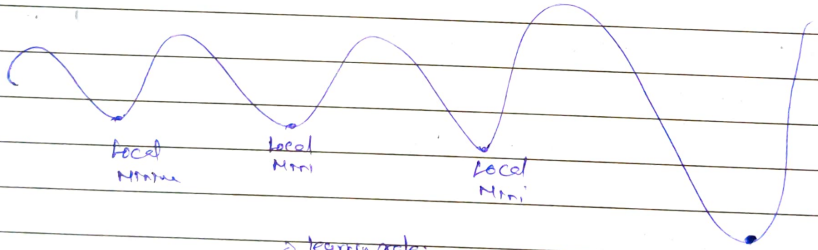




neither concave nor ~~the~~ convex



* GD help to find ~~the~~ to find minima.
GD doesn't know it is easy to find minima or maxima.
* Gradient ascent → gives maxima
→ Min - f(x)



learning rate
$w_{t+1} \leftarrow w_t - \eta \frac{d}{dx}(L(x))$

* Linear regression is a convex curve
* Based on quality metric we can find whether it is curve is convex and non-convex.
* Logistic regression used for type of classification.

OLS → Approach
MLE → Based Approach

M T W T F S S
Page No.:
Date:
YOUVA

* Statsmodel is well for linear regression model but not so for logistic regression as it is bit hard to implement hence, Sklearn is used.

* Linear regres → Close

$$W^* = (X^T X)^{-1} X^T y$$

Inverse = $O(n^3)$

5. Difference bet ~~disadvantage~~ of Min-max over Standardisation.

M T W T F S S
Page No.:
Date: 02/1/22
YOUVA

→ Linear regression is used in the various field such as real estate, telecom, e-commerce etc to build predictive model.

→ We use mean, median, mode to 'impute' missing value.

→ The main method to visualise Categorical variable and continuous variable is Boxplot.

→ '16 Convert a Column from series to dataframe enclose it with in '[]'.

→ For Categorial Variable "mode" is used to impute.

Rescalling

1. To bring the coefficient of different variable in a certain range by bring all the variable in a comparible scal
2. (Interpretibility) (Interpretability)
2. If we rescale the variable within the range of 0-1 then the optimisation happens behind the schene becomes much fast i.e minimisation routine becomes much more fast when we train a network using gradient descent funtn.
3. Min-Max rescalling should be used more as it take care of outlier present in the ~~data~~ input by putting the outlier point to 1 and non-outliers in beto 0-1.

Note:- The Advantage of Standardisation over the other is that it doesn't compress the data beto a particular range as in Min-Max scaling. It is useful when there are extreme data point (outlier)

→ SKlearn comes with a package called "Preprocessing" which has MinMaxscaler ifuntn in it.

→ Fit on data

fit: learns xmin, xmax on train data set.

transform(): Calculates (x-xmin)/(xmax-xmin)

fit_transform(): does the work of above 2 method in one method

3. Training the Model

→ Plot heat map to check the correlation between all the numeric variable.

→ Check the variable which has high correlation then keep on adding other called "Bottom-up" Approach

Note: Even 2 to 3% increase in the value of R-Squared value after adding a variable to the model means a Significant change.

Method to Drop Variable

(P) → P(t) below 0.05 (low) and vice-versa

(2) → Significance (p-value)

(2) VIF

- High P-value, high VIF (Drop that variable)
- High p-value, low VIF (Remove these first)
- Low P-value, high VIF (Remove these after the one above)
- low-P-value, low VIF (keep it)

Note: Drop a variable one by one, because after droping the variable with highest VIF the other high VIF's variable will also get effect due to the drop of ~~VIF~~ variable with highest VIF.

→ Never fit on Test Sets. because test set are unseen data for the model so we don't want the model to know about it (i.e its mean, max).

ML C34 - Linear Regression Module

Speaker -> Karan

Residual

$$RSS = \sum_{i=1}^{N} (\text{Actual-output} - \text{predicted-output})^2$$

~~Total~~ Square Sum

$$TSS = \sum_{i=1}^{N} (\text{Actual-output} - \text{average-of-actual-output})^2$$

It tells how much the data point move around the mean.

$$R^2 = \frac{\Sigma (y_p - \bar{y})^2}{\Sigma (y - \bar{y})^2}$$

$$RSE = \text{Square root} (RSS / \text{degree of freedom})$$

F-Statistics -> It tell relation the predictable model with the a model with all β coefficient with 0 value.

F-Statistics Prob -> It tells the probability of a model with all β coefficient is equal to 0.

$$H_0 = \beta_1 = \beta_2 \cdots \beta_n = 0$$

F-Statistics:- If your assume all β coefficient is 0 then how good we are doing from that model.

-> It tell the model which we have created and

Linear Regression

1. Process of ~~estimate~~ estimating relationship betn variables.
2. Explains change in dependent variable with change in the values of predictors:
   - Simple linear Regression: Changing only one variable at a time
   - Multiple linear Regression: Changing multiple variable at a time.
3. Uses :- Forecasting and Prediction
   → Regression guarantees "interpolation" of data not "extrapolation" necessarily.
4. Show correlation, not causation
   - Correlation: doesn't imply causation.
5. A form of Parametric Regression

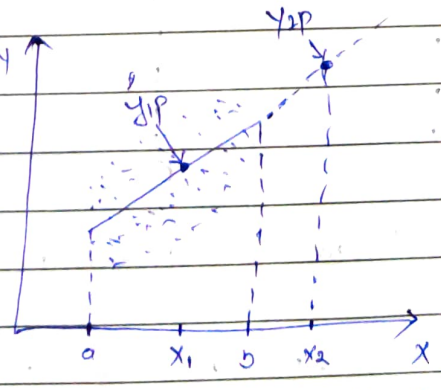
Parametric Regression	Non-parametric Regression
1. Data follows fixed parameters	1. Data doesn't follow fixed parameter
	2. Dynamic in nature.

Interpolation :-

It basically mean using the model to predict the value of dependent variables on independent values that lies within the range of data we already have.

Extrapolation :-

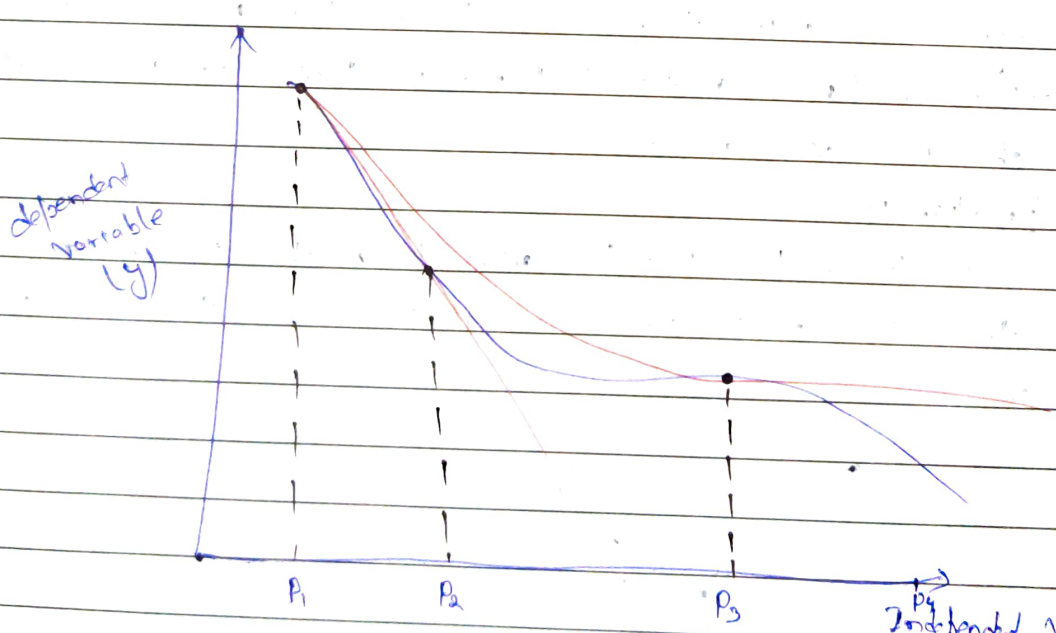
It means predicting the dependent variable on the independent values that lie outside the range of the data the model was built on.



Y_{1p} = Interpolation ; Y_{2p} = extrapolation

Prediction vs Projection (Forecasting)

	Prediction	Projection
Importance of Outcome	Focus: Identifying the driver variable and measuring their impact on dependent variables.	Focus: Final projection result / forecasted value
Assumption	No specific assumption is considered	Assumes everything remains the same as today. Forecast will change if new incident occurs.
Complexity / Accuracy of Model	Simple models are better than complex models.	Choose accuracy over explanation.



1. Model made using P_1 and P_2 we can't predict accurately the value betn P_2 and P_3.
2. Model made using P_1 and P_3 we can't predict accurately betn P_3 & P_4. (extrapolation)

Assessing the Model Stability

→ If R-Squared and Adjusted R-Squared are very close it means ~~adjusted~~ none of the parameter are redundent in the model means no extra variable is added to the model.

→ Errors bet actual and predicted O/P is normally distributed which confirms that there are no variables the could have helped to explain the model better.

1. Data set is small
   - Not advisable to separate testing and training sets.
2. Use bootstrapping
   - Choose 10-20% of sample randomly
   - Opt for multiple iterations with replacement
   - Train model on "in-sample" group
   - Test on "out-sample" group.
3. Model is stable when R-squared is similar

Note:- Least Square Error

① Used to find the best fitted line through the set of points.

Mean Square Error

1. Used to evaluate model after fitting it.
2. It finds out the average of the difference bet'n the actual and predicted value is a good parameter to compare various models on the same data set.

Correlation Coefficient

→ ① Pearson's R (used for linear relationship b/w variable)
→ ② Spearman's R (used to determine the correlation bet'n non-linear variable)

Bias

→ Bias mean difference bet'n predicted value by model and the real values.
→ It is a Error and need to have low bias for ML model.

Variance

→ Sensitivity of the model to small fluctuations in the training dataset.
→ Low variance for ML algorithm.

Note:- - Straight line mode will have low variance, high bias
- High degree polynomial will have low bias, high variance

SGL-ML-C34-CC-G04

AI/ML Research	Data Scientist	ML Engineers
Computer Vision Engineer		
Research & Develop	Implement	Large Scale deployment

(i) Applied team :- Analyses the given model and check it feasibility, it or not

(Data Engineers + ML Engineer)

(1.1) Data Scientist ⇒ EDA, Decide/rank to use ML is required or not

(1.2) ML Engineers Build ML model

* Data Scientist
  ↳ EDA

* ML Engineer
  ⇒ Build DL, ML model

Google Dataset