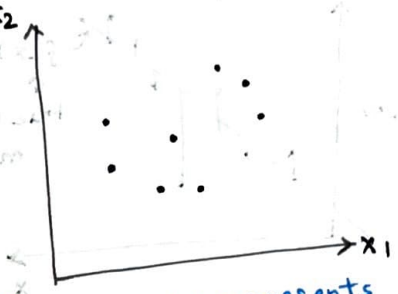
For a customer segmentation problem, Euclidean distance helps us to group and differentiate light buyers from heavy buyers. While the cosine similarity helps to group customer with respect to the similarities in terms of their buying habits and tastes.

Distance matrix: which customers are more similar? Diagnol elements - 0

Similarity matrix: Correlation matrix (cosine similarity) Diagnol elements are all 1, hence

x_1 → no. of units of product A that has been bought

x_2 → no. of units of product B that has been purchased



Each dot represents a customer.

This framework won't allow us to handle new customers.

Single linkage: distance between 2 clusters is minimum of the all possible distances between 2 points.

Complete: max

Average: Mean of pairwise distance between points in 2 different clusters

Dendogram → used to visualize the number of clusters.

→ Observations are along x axis and distance along y-axis.

Cutting the dendogram →

Within cluster → homogenous collection of observations.

Between clusters → heterogenous collection of clusters

The main objective of cluster analysis is to ~~create clusters~~ find groups within the data such that similar elements are present inside a group and very different elements are present in different groups. Finding groups within a data helps us to benefit economically and increase action effectiveness.

Targeting potential customers: conversion rate is high, time and economically beneficial.

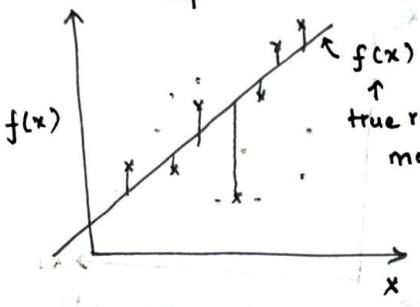
31/10/2022

SOURCES OF ERROR: (Most ☆ conceptual topic).

ERRORS

- Irreducible Error → cannot be minimized.
- Reducible Error

Population

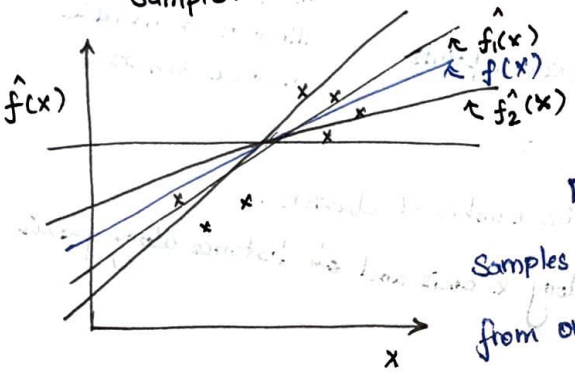


σ_ε^2 = var(residuals) = irreducible error

→ obtained by fitting the population ~~data on~~ linear regression model on population data.

→ focus on errors that are reducible.

Sample.



$\hat{f}_1(x)$ = LR model fitted on the sample s_1

$\hat{f}_2(x)$ = LR model fitted on the sample s_2

For different samples, we have different samples since y-intercept and slope changes from one to another sample.

→ The sample mean is a function of the sample we are choosing.

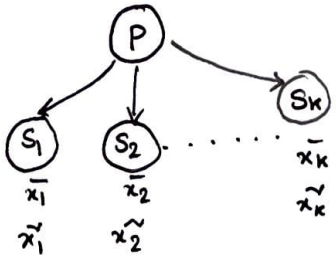
→ When the sample size is smaller the variability of sample means (the sample mean vary alot from one sample to another).

① A salesman who quotes 15 Rs, 20 Rs, 25 Rs, 30 Rs for a marker.

② Another salesman quotes Rs 18, Rs. 19, Rs. 20, Rs. 21 for a marker.

Actual marker price = Rs. 20.

② is more reliable because $SD_2 < SD_1$.



$sd(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k)$

$sd(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_k)$

$Sd(s_{d_1}, s_{d_2}, \ldots, s_{d_k})$

} STANDARD ERROR ↑ sd of the sample statistics.

Standard error determines or measures the reliability of the estimate.

~~It is a range~~

95% confidence interval of μ.

Sample mean = 10

$SE(\bar{x}) = 2$

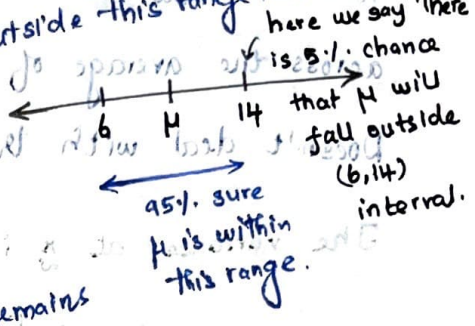
$10 \pm 1.96 \times 2$

$\Rightarrow$ (6, 14) $\Rightarrow$ 95% CI estimate of μ.

We are 95% sure that pop. mean will lie between 6 and 14.

$\hookrightarrow$ $P(\mu \text{ is in } (6,14)) = 0.95$; $\times$ cannot be true.

↳ here μ has 5% chance lying outside this range. We're here saying μ is moving.



So we rewrite this as,

$P(\text{interval contains } \mu) = 0.95$

↳ here the interval can move while μ remains fixed.

If the average of all the regression lines differs alot from the true regression line, then there's a problem.

How far is the average of all the models is away from true model is called the bias.

But in real life, we don't have the true regression line.

Two sources of reducible error:

(1) Bias

(2) Variance

Let $f(\underset{\sim}{x})$ represent the true regression model (ii) model fitted on the population data.

Let $\hat{f}_i(\underset{\sim}{x})$ be the fitted regression model on the ith sample and $\bar{f}(\underset{\sim}{x})$ be the average of all the fitted models (ii) (the average of all the fitted values from all possible fitted models for the input $\underset{\sim}{x}$.

The bias of the model at $\underset{\sim}{x}$ is defined as:

$$\text{Bias}(\underset{\sim}{x}) = f(\underset{\sim}{x}) - \bar{f}(\underset{\sim}{x})$$

Bias measures how far is the average of all possible fitted models is away from the true model.

→ If this is large then we got a problems. We deal with the predicted values.

Variance → how much are the individually fitted models varying across the average of the fitted models.

Doesn't deal with true model at all.

The variance at $\underset{\sim}{x}$ is defined as: predicted value made by all models on an average (or) average regression line.

$$\text{var}(\underset{\sim}{x}) = E\left[\left(\hat{f}_i(\underset{\sim}{x}) - \bar{f}(\underset{\sim}{x}) \right)^2 \right]$$

→ Variance measures how much does the outputs of each models (fitted over all possible samples of the same size) varies from the average of the models.

The mean squared error at $\underset{\sim}{x}$ (one particular input) is,

$$\text{MSE}(\underset{\sim}{x}) = \text{Irreducible error}(\underset{\sim}{x}) + \text{bias}^2(\underset{\sim}{x}) + \text{variance}(\underset{\sim}{x})$$

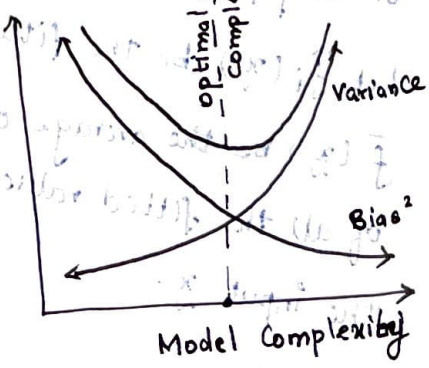
Bias is dependent on one particular value of x and is not same for all x.

The model should not look different for different samples of same size.

for decision tree, model complexity is dependent on the depth of the decision tree.

MODEL COMPLEXITY - BIAS - VARIANCE.

Feature selection is quite ... to determine the point @ which the Total Error



model learns significantly less → underfitting
Model learns sufficiently large → overfitting.

High variance will make the model overfit the data. While high bias makes the model underfit the data. Both are bad.

A KNN model with K=2, K=20 → smoother model
↓
agitated model.

Book: ISLR.

Lesser the value of K → overfitting the model
Higher the value of K → underfitting the model.

Always work with samples of larger size.

Model complexity ? for kNN.

Exp 1: Let P = cars data with 406 observations and the variables MPG, weight, horsepower and displacement. Generate 100 samples of size 100 each from P, over each sample, fit a decision tree model to predict MPG of depth = 2 and a decision tree model of depth = 8. For a fixed input, get the predicted values for both the kinds of models.
Compute the variance of the predicted values seperately for
① DT (depth = 2) ② DT (depth = 8). Report your observations and comment.

Exp 2: The minimum no. of observations at leaf node controls the complexity of a decision tree. For each sample fit a decision tree model with minimum_samples_leaf = 5 vs. min_samples_leaf = 20. For a fixed input compute the variance of the predicted values seperately.
↙ if set to large value model complexity is reduced

Exp 3: Using similar experiment as above compute the variance of a 2NN model and a 10NN model.

Print out your CV's.

Books
↓
Larose and Larose ←
ISLR

→ Take notes on AUC-ROC curve given to Senior batch Students.
→ If necessary talk with Anchal Khanna.
→ Notes on data leakage.
→ Final Assignment on Graphs, Whatsapp Assignment on
→ Decision Tree Assignment on House Price prediction

→ t-test in Linear regression ? why not z-test ?

→ VIF → ?

VIF: Variance Inflation factor.

$R^2 \rightarrow 0.80 \rightarrow$ 80% of the variance of Y is explained by X using this model.

variance → measure of amount of information present in a data

2, 2, 2, 2, 2 ← values are same throughout

12, 5, 32, 105, −18 ← gives more information

X	Y

$y_i = b_0 + b_1 x_i + \varepsilon_i, \quad i = 1(1) n$

$\hat{y} = \hat{b_0} + \hat{b_1} x$

$\hat{b_0}, \hat{b_1}$ } Estimated values given by model.

For multivariate settings, when you have 2 or more predictors and a target variable, Y, we can't use correlation. In such cases we go with multiple correlation.

$cor[Y, [x_1, x_2]]$ ← measures linear dependency of Y on both x_1 and x_2.

Pearsons coefficient → can be used to express the linear relationship between 2 variables.

To determine multiple correlation we need to set a linear relation such that:

$cor(Y, b_0 + b_1 x_1 + b_2 x_2) = cor(Y, \hat{Y})$

Squared of multiple correlation → $R^2 = cor(Y, \hat{Y})^2$

mult. corr $(x_1, \ldots)$ ← correlation of x_1 with all other variables → single value (−1 to +1)

If near to 1, x_1 is highly correlated to all other columns.

$cor(x_1, b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_{60} x_{60})$ → single value.

$cor(x_2, \ldots\ldots\ldots)^2$ → single value.

→ For simple regression model, $r^2 = R^2$ (Pearsons coefficient)

→ Multiple correlation → measure multicollinearity.

VIF:

→ Remove redundancy, if one variable is explainable by any 2 variables, then we can remove the 1st variable.

VIF → doesn't measure multicollinearity.

Does VIF measure multicollinearity ? ← AIM magazine.

Multicollinearity → Multicorrelation if high, we drop the column.

VIF:

	Predictor	Target	Predictor	
①	X_1	~	$X_2 + X_3$	$R_1^2 = 0.90$ → 90% of var. of X_1 is explained by X_2 and X_3
②	X_2	~	$X_1 + X_3$	$R_2^2 =$ ____
③	X_3	~	$X_1 + X_2$	$R_3^2 =$ ____

Find individual R^2 assuming one variable as target and rest as predictors.

VIF = 1, the variable is uncorrelated with all other variables.

$$VIF = \frac{1}{1-R^2} \Rightarrow$$

① $VIF_1 = \frac{1}{1-0.9} = 10$

② $VIF_2 = \frac{1}{1-0.8} = 5$

③ $VIF_3 = \frac{1}{1-R^2}$

$0 < R^2 < 1$

$1 < VIF < \infty$

If VIF is high, we drop the particular column (but benchmark should be set)

Using VIF to reduce multicollinearity:

Step 1: Fit a model with all the predictors.

Step 2: Calculate the VIF of all the predictors in the model.

Step 3: Let O_1 be the VIF benchmark. Drop the predictor with the highest VIF provided it exceeds O_1.

max. VIF

Note: Only drop that predictor provided it's greater than O_1. After dropping the highest VIF predictor, the VIF of all predictors will significantly fall.

Step 4: Refit the model by dropping the predictor (with all the other predictors).

Step 5: Repeat step 2-4 until all the VIF's of the selected variables

Step 3: Divide into train-test-validation-split

(or) use cross validation.

Step 4: Run EDA tools (create a package of your own):

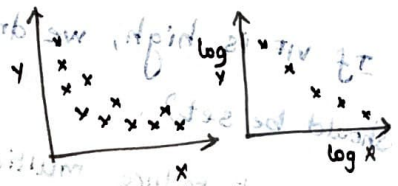
Create a package that will help you generate boxplot, histogram for univariate, for bivariate (numerical ~~predictors~~ target and numeric variables.

categorical predictors → side by side boxplot).

→ A predictor that's highly collinear with target should be kept.

→ Transformation like log (to increase linearity, shrinkage); square root transformation (u) a longitudinal relationship starts to look like linear.

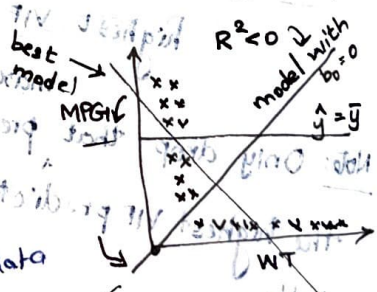
We take log on both variables.



Step 5: Fit a linear regression model based on all the X (features); ← baseline model.

Can R^2 be negative?

A model which performs worse than mean model then R^2 can be negative, unless otherwise we're manipulating or failed to fit the intercept



OLS regression → R^2 can never be negative.

R^2 → how well model has fitted on training data

→ measure of goodness of fit, only to access fit of model.

R^2 → should not be measured on test data.

To measure error metric go with MAE, RMSE and MAPE not R^2 on test data.

→ If R^2 is high, overfitting occurs. If you drop variables, overfitting reduced.

R^2 will fall, validation performance will increase

Adjusted R^2:

No matter what you add a feature, R^2 will increase. So we add some penalty; (i.e) penalized R^2 ≈ adjusted R^2.

→ Adding relevant predictors will lead to increase adjusted R^2. Adding features may or may not increase the adjusted R^2.

→ Adjusted R^2 can be used for feature selection

Can adjusted R^2 be used to measure error on test data?

Checks for multicollinearity → to drop columns (VIF)

Engineering Features:

→ Generating new features from existing features. To improve model accuracy and decrease in error rate.

Forward selection: which features can finally go into our model.

↳ uses adjusted R^2 to figure out which variable can go into the data.

① x_1, x_2, x_3, x_4 ← Predictors.

$y \sim x_1$, Adj R^2

$y \sim x_2$, Adj R^2

$y \sim x_3$, Adj R^2 ← highest value, then x_3 is the first predictor that goes into model.

$y \sim x_4$, Adj R^2

② $y \sim x_3 + x_1$, A.R^2

$y \sim x_3 + x_2$, A.R^2

$y \sim x_3 + x_4$, A.R^2 ↑ highest ⇒ x_4 is the next selected feature goes into model

③ $y \sim x_3 + x_4 + x_1$, AR^2 ↓ lower than previous

$y \sim x_3 + x_4 + x_2$, AR^2 ↓ none.

Then we stop. We consider only x_3 and x_4 as my final features.

Why have you selected this model? Objective is reducing the error, go for whichever model gives best results.

→ For every instance of test data, KNN goes through each and every observation on test data. for eg test data (100) training data (100000) Then for loop runs for $100 * 100000$. So KNN is lazy learner while Decision tree is greedy learner.

→ Interpretation → Linear regression (or) DT (but bushy tree is not easily interpretable).

→ Even before fitting model, in linear regression we assume there exists a linear relation between the target and predictor.

→ DecisionTree Regressor → automatically figures out things and splitting

→ If we're really clear about x and y, parametric model like linear regression performs better than non-parametric model (like decision tree).

→ If linear regression gives lesser accuracy, we failed with their assumption or hypothesis.

KNN vs Linear regression → ISLR

non-parametric parametric model.

When you will use linear ~~relatio~~ regression or decision tree?

How will you choose between LR or decision tree?

→ In higher dimensional space, the plane is a flat surface and looks like straight line in n-d plane whereas in 2D it can be a curve.

→ Everytime we try to fit the model we'll be making an assumption.

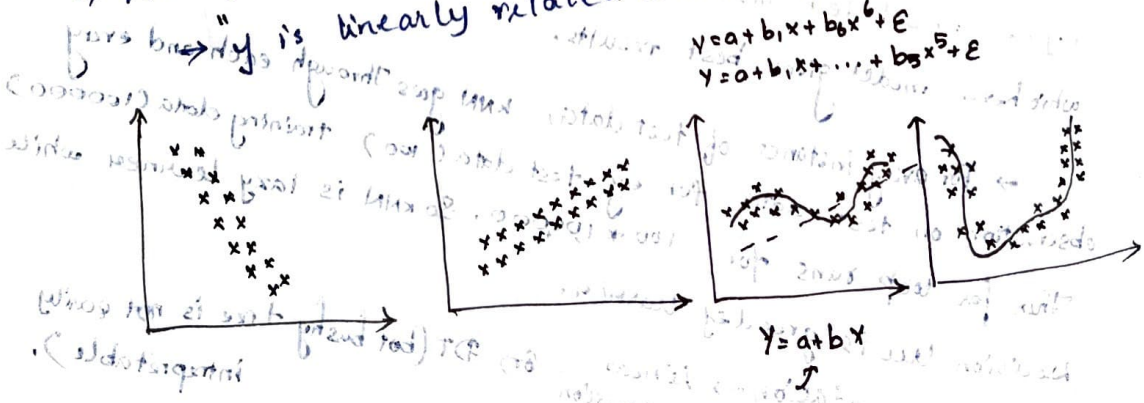
$$y = a + bx + \epsilon$$

→ We'll use assumption to fit linear regression.

The 1st assumption we're making is, y is in linear relation with features.

→ Assumption 1 cannot be checked.

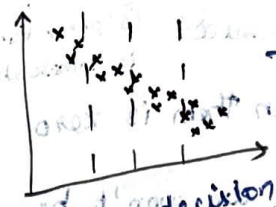
→ "y is linearly related with coefficients"



→ The assumption you made is wrong, linear regression doesn't fail.

→ Use linear regression only after EDA after you get clear picture between target and feature.

Decision Tree: splitting happens on feature space based on homogeneity of target variable.



→ In LR, we need to prove hypothesis (which explain the target with predictor in best possible way).

→ We make decision tree to find patterns instead of we searching patterns.

→ Linear regression involves lot of manual searching.

Why not use decision tree all the time?

→ A LR model is a parametric model and parametric models have high predictive power. If I know x and y well, a linear regression outperforms the decision tree.

→ Parametric model outperforms non-parametric model

To achieve linearity we can use logarithmic transformation, higher order transformation.

DT → no assumptions made with parameters.

Never compare the RMSE of $\hat{y}$ with RMSE of $\log \hat{y}$, we must take antilog to find right RMSE.

RMSE 1000 ← $y = a + bx + \varepsilon$

RMSE 4.5 ← $\log(y) = a + b\log(x)$ "ideal" homoscedacity

→ Always suspect the too good things.

→ Increasing the power of the polynomial (or order of features) increasing the model complexity and leads to overfitting.

Regularisation?

→ All categorical variables are considered one (all under one dummy).

All dummy variables should be looked as same variable (should not be looked as different columns, values should be looked as vectors).

$MPG = b_0 + b_1 O_{UK} + b_2 O_{US} + \varepsilon$

(0, 0) ← Japan
(1, 0) ← US
(0, 1) ← UK

MPG	Origin	O_{US}	O_{UK}	
	US	(1	0)	← vector rep US
	UK	(0	1)	← UK
	Jap	(0	0)	← Reference level

When origin changes from Japan to UK, the MPG ↑ by b_1 units.

① $y_i = b_0 + b_1 x_i + \varepsilon_i$, $i = 1(1)n$ ← random variable

y is linearly related with coefficients.

$\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, $\forall i = 1(1)n$ ← mean

② i = independently ↔ covariance between x_1, x_2, $cov(\varepsilon_i, \varepsilon_j) = 0, \forall i \neq j$ ← errors should not be autocorrelated.

③ i = identically → Normal distribution with same mean and variance.

④ d = distributed ④ Errors are normally distributed ③ error variance is homoscedasticity

For any pairs of errors, the covariance between them is zero.

identically → no matter what error I bring, we won't be able to distinguish between distribution because all errors follow normal distribution with same mean and variance.

Homoscedacity → Variance of errors are constant

Heteroscedacity → error variance are not constant

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

→ If data has one column which can be represented as a linear combination of other 2, then we have a dependent column.

Considering as matrix $\begin{bmatrix} 2 & 6 & 0 \\ 5 & 15 & 1 \\ 8 & 24 & 0 \end{bmatrix}$ ← singular matrix $\Delta = 0$

No need to use $(X^T X)^{-1}$ if we use gradient descent

$\hat{\beta} = (X^T X)^{-1} X^T y$ ← inverse is not possible since the feature matrix is not invertible.

→ So we need to remove strong multicollinearity so that we get pretty good results.

→ Complete case analysis: A case is in observation or record.

Handling Missing values:

A case that doesn't have any missing values in it

Complete Case Analysis: Restrict the analysis to only those cases which are complete. Delete all cases which contains atleast one missing values in it. Only if it is legitimate.

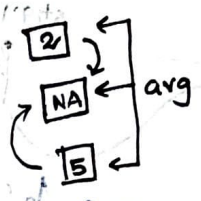
Caution: This may lead to a massive loss of information.

If we're randomly sample, the sample we pick is not random if we're removing observations resulting in more bias.

Never drop observations.

For sequential data, we need to look into the frequency (like time series) we can use forward and backward fill.

→ Do not replace missing values with some arbitary values.



→ Incase of timeseries, the timestamps should be of same interval.

→ If distribution is very skewed we go with median imputation.

→ Mean imputation makes a lot of sense because it's the expected value of the random variable.

→ If there's a lot of missing values, replacing missing values with