Explainable AI

1) Global - Explainability
2) Local Explainability
3) Post-hoc Explainability
4) Intrinsic -

1) Global - overall explainability
   - once we build the model with help of the final model parameter we explain the why?
   - example - Linier - $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \cdots$

   where $\beta_0, \beta_1, \beta_2$ are the coeffic
   get calculated & we explain what are the values/weighted of thoes feature to get final value.

2) Local explainability
   - with the single row if we are predicting the output then we explain why per that model
   like x_1, x_2, x_3 are feature which produces value y_1 then why it & how get that value.

Post-ahoc →
1) after training done we get some values so we need to use SHAP or LIME to explain the why?

Intrinsic → with help of this we can explain the features those play important role in calculate output value.
- linear — we use coefficent
Random forest and decision tree we ~~use~~ calculate important feature.

Real world use cases
1) Banking - credit scoring, fraud detection
2) Healthcare - Patient Risk scoring
3) Retail - Recommendation
4) Public sector - risk mgt/assessment resource allocation

challenges in model explainability
- Accuracy vs Interpretability
more accurate model harder to explain
- Domain knowlege req.
- Communicating to non tech audience
- Risk of misinterpretation.

Explainability builds trust, ensure fairness meets regulations & improve model.

SHAP- Shaply, additive explaination.

Dataset → avg prediction

Base value → $\phi = \frac{1}{n} \sum_{i=1}^{n} f(x_i)$

$\hat{y}$ = Base value + sum (shaply value)

$\hat{y} = \phi_0 + \sum_{i=1}^{d} \phi_i$

How it works →

1) calculate Base value
2) calculate final prediction
3) check out additivity

$\hat{y} = \phi_0 + \phi_1 + \phi_2 + \cdots$

ϕ_1 ↑ x_1 , ϕ_2 ↑ x_2 ← feature.

Component →

- model specific – optimized
- model agnostic – no care which model we use.

1) **Explainers** → → calculate SHAP value
→ output value of explainer called explanation

→ Explainer are brain of SHAP - calculate SHAP value.

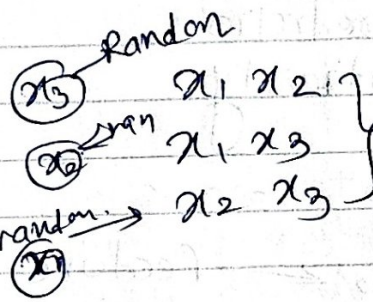
- Tree Explainer –
- kernel Explainer – slow
- Linear Explainer –
- Deep Explainer –

2) maskers →

→ control how SHAP hides or replace features when calculating their contribution

- They has background dataset

- suppose we have three feature $x_1 x_2 x_3$ & we are calculating $\hat{y}$ if we want to get contribution of x_1, x_2 without x_3 we can either replace with random value or constant or avg value so that doesn't matter while calculating $\hat{y}$.



3) Plot →

- Summary plot
- waterfall plot
- Dependance
- Decision plot

4) Background data → Reference Point

when SHAP calculates the contribution of of feature it need to simulate - 'what would the prediction be if we didn't know feature value'.

Dint know - leave it blank. but model can't handle it. SHAP replace with some reference point. The reference value come from Background data

Example →

f_1 = Age

f_2 = Income

Age	Income
30	60
40	80
35	70

prediction = 10 + 0.5 Age + 0.1 Inc

$P_1 = 10 + 0.5 \times 30 + 0.1 \times 60 = 31$

$P_2 = 10 + 0.5 \times 40 + 0.1 \times 80 = 38$

$P_3 = 10 + 0.5 \times 35 + 0.1 \times 70 = 34.5$

$\hat{y}_{test} = 10 + 0.5(50) + 0.1(200) = 45$

$\hat{y}_{test} = \phi_1 + \phi_{age} + \phi_{income}$

$45 = \text{basevalue} + \text{Age contribution} + \text{income contri.}$

Calculate base

$\phi_1 =$ Average prediction of values.

$= \frac{31 + 38 + 34.5}{3} = \frac{103.5}{3}$

$= 34.5$ (basevalue)

what the base value mean → if we dont have any feature value so our model provide prediction as base value.

(2) calculate contribution of Age column.

Age = 30 + 40 + 35 / 3 = 105/3 = 35

New dataset

Age	Income
35	60
35	80
35	70

→ model → $\hat{y}$

$y_1 = 10 + 0.5(35) + 0.1(100)$

$= 10 + 17.5 + 10$

$= \underline{37.5}$

ϕ_{age} = Final prediction − prediction without age

$= 45 - 37.5$

$\boxed{\phi_{age} = +7.5}$

$\phi_{income} = 60+80+70/3 = 210/3 = 70$

$y_{pred} = 10 + 0.5(\cancel{350}) + 0.1 \times 70$

$= 10 + \cancel{187.5} + 7$

$= \cancel{350}\ \cancel{205}\ 42$

ϕ_{income} = Final prediction − prediction without income

$= \cancel{37.5}$

$= 45 - \cancel{37.5}\ 42$

$\phi_{income} = \cancel{+7.5}\ 3$

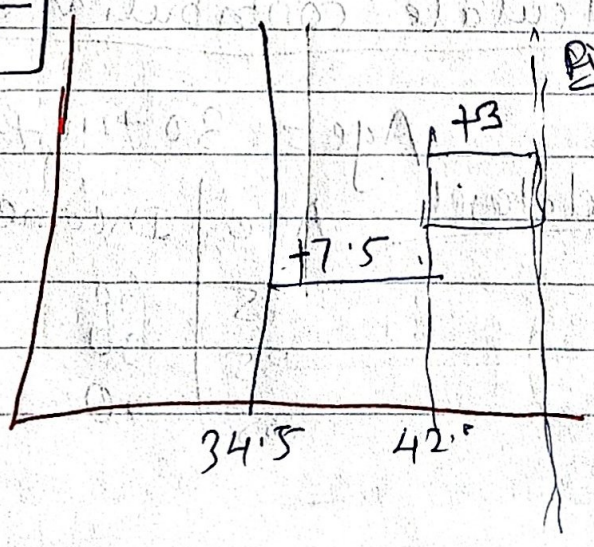
$\boxed{\phi_{income} = 3}$

age contribution

income contribution

$\hat{y} = 34.5 + 7.5 + 3$

$\boxed{\hat{y}_{test} = 45}$



SHAP – Shaply additive Explaination

Summary –

1) calculate base value – Avg of all pred
2) calculate contribution of features
3) check which feature contribution is more.

How to calculate shaply value

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} (v(S \cup \{i\}) - v(S))$$

$$\sum_{i \in N} \phi_i(v) = v(N) \leftarrow \text{check for additivity}$$

ϕ_i = indivial contribution of feature.
$v(N)$ – final prediction.

Game theory → example kaggle compition.

suppose we have 3 player P1, P2, P3 who entered in the kaggle compition. they all have their strength & whenever they team up they win 1st price. so Now suppose they win $90000 then with our normal calculation everyone get $30000 but this is not shaply does SHAP lr calculate the contribution of each player/value in the compitetion so thats the shaply value concept so players are feature and shap values are contribution of each player.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left\{ v(S \cup \{i\}) - v(S) \right\}$$

$|N|$ → length of set → no of feature/player.
$|N| \rightarrow 3!$ (in this example)

S → subset $\{P_1 P_2\}$ ← example.
$|S|$ → length of subset.
$v(S) = F(S)$ → model or mathamatical function or prediction.

$v(\{P_1, P_2\}) = \$5000k$ — outcome / pay of value.

~~$v(S)$~~ $v(S \cup \{i\}) \rightarrow f(x) \rightarrow \$9000k$ in this example

S ← subset of $(P_1 P_2)$
$\{P_3\}$ ← in this example

$v(S \cup \{i\}) - v(S) = \{P_1 P_2\} \cup \{P_3\} - \{P_1 P_2\}$
= will get individual contribution of P_3

$\sum_{S \subseteq N \setminus \{i\}}$ = calculation of summation of subsets where i is not part of it.

$N = \{P_1, P_2, P_3\}$
subset S →
$S_1 = \{P_1 P_2\}$ or $\{P_2 P_1\}$ if $i = P_3$
$S_2 = \{\}$
$S_3 = \{P_1\}$
$S_4 = \{P_2\}$

here S is 4

$$= \sum_{A \subseteq N / \{P_3\}} \frac{|2|!\,(3-2-1)!}{3!} \left(\{P_1 P_2 P_3\} - \{P_1 P_2\} \right)$$

*Example – 2 player Problem.

player A
player B
} 10

1) $v(\phi) = 0$
2) $v(\{A\}) = 4$
3) $v(\{B\}) = 6$
4) $v(\{A,B\}) = 10$

} given.

$\phi_A \rightarrow$ shapley values $\rightarrow$ for A

$$\phi_n = \frac{|S|!\,(N-S-1)!}{|N|!}\left(v(S\cup\{i\}) - v(S)\right)$$

empty set $S_1 = \phi$. $S = 0$, $N = 2$

$$= \frac{0!\,(2-0-1)!}{2!} = \frac{1\times 1}{2} = \frac{1}{2}$$

$$= \frac{1}{2}\left(v(S\cup\{i\}) - v(\phi)\right)$$

$$= \frac{1}{2}\left(v(S\cup\{i\}) - 0\right)$$

$$= \frac{1}{2}\left(v(\phi\cup\{A\})\right) = \frac{1}{2}\times 4 = 2$$

$S_2 = \{B\}$

$$= \frac{1!\,(2-1-1)}{2!} = \frac{1}{2}$$

$$= \frac{1}{2}\left(v(S\cup\{i\}) - v(\{B\})\right)$$

$$= \frac{1}{2}\left(v(\{B\}\cup\{A\}) - 6\right)$$

$$= \frac{1}{2}(10-6) = \frac{1}{2}\times 4 = 2$$

$\phi_A = \phi_0 + \phi$
$= S_1 + S$
$= 2 +$
$= 4$

same

Player B $= \left(\frac{1}{2}\right) \cdot \{V(S \cup \{i\}) - V(S)\}$

$S_1 = \phi$ (emptyset)

$= \frac{1}{2} \cdot \{\{\phi\} \cup \{B\} - V\{\phi\}\}$

$= \frac{1}{2} \cdot (V\{B\} - V\{\phi\})$

$= \frac{1}{2} \times (6 - 0) = \frac{1}{2} \times 6 = 3$

$S_2 = \phi_A$

$= \frac{1}{2} \times (\{B\} \cup \{A\}) - V\{A\}$

$= \frac{1}{2} \times (10 - 4) = \frac{1}{2} \times 6 = 3$

$\phi_B = \phi_0 + \phi_A = 3 + 3 = 6$

$\phi_A = 4 \quad \phi_B = 6$

additivity rule $= \Sigma V(\phi_i)$

$V(\{A, B\}) = V(\phi_A) + V(\phi_B)$

$10 = 4 + 6$ ✓

3 Player Problem

$(P_1 P_2 P_3)$ – Player

$\{P_1\}$ · 6
$\{P_2\}$ 12
$\{P_3\}$ 6
$\{P_1 P_2\}$ 24
$\{P_1, P_3\}$ 18
$\{P_2 P_3\}$ 18
$\{P_1, P_2 P_3\}$ 30

$\phi P_1 \ \phi P_2 \ \phi P_3 = ?$

$\phi P_1 = ?$ = shaply value = contribution of P_1

subset length

make subset for $P_1 \rightarrow$ $\{P_2\} = 1$
$\{P_3\} = 1$
$\{P_2 P_3\} = 2$
$\{\} = 0$

For P_2 – $\{P_1\} = 1$
$\{P_3\} = 1$
$\{P_1 P_3\} = 2$
$\{\} = 0$

For P_3 = $\{\} = 0$
$\{P_1\} = 1$
$\{P_2\} = 1$
$\{P_2 P_3\} = 2$

formula = $\sum$ weight × (with feature – without fe

$|N| = 3$

$$\text{weight} = \frac{s!(|N|-|s|-1)!}{|N|!}$$

$$\text{weight for } s_1 = \frac{0!(3-0-1)!}{3!} = \frac{1(2!)}{3!} = \frac{2}{3\times 2} = \frac{1}{3}$$

(s_1: empty)

$$\rightarrow s_2 \text{ for } P_2 = \frac{1!(3-1-1)!}{3!} = \frac{1}{3} = \frac{1}{6}$$

$$\rightarrow s_3 \text{ for } P_3 = \frac{1}{6}$$

$$\rightarrow s_4 \text{ for } (P_2 P_3) = \frac{2!(3-2-1)!}{3!} = \frac{2!\times 0!}{3!} = \frac{2}{3\times 2} = \frac{1}{3}$$

$$\rightarrow s_5 \text{ for } (P_1 P_2) = \frac{2!(3-2-1)!}{3!} = \frac{2!\times 0}{3!} = \frac{1}{3}$$

$$- s_6 \text{ for } (P_1 P_3) = \frac{2!(3-2-1)!}{3!} = \frac{1}{3}$$

$\Phi P_1 = \{\}, \{P_2\}, \{P_3\}, \{P_2 P_3\}$

	outcome (S)	v(S ∪ {P1})	diff.	
P1				
ϕ	0	6	6	2
{P2}	12	24	12	2
{P3}	6	18	12	
{P2 P3}	18	30	12	4

$\phi P_1 = 2+2+2+4 = 10$

ϕP_2

P2	outcome	v(S ∪ {P2})	diff	weigh	out
ϕ	0	~~18~~	12	1/3	4
{P1}	6	24	~~18~~	1/6	~~3~~
{P3}	6	18	12	1/6	2
{P1 P3}	18	30	12	1/3	4

ϕP_3

P3	outcome	v(S ∪ {P3})	diff	wei	dif
ϕ	0	6	~~12~~ 6	1/3	2
{P2}	12	18	6	1/6	1
{P1}	6	18	12	1/6	2
{P1 P2}	24	30	6	1/3	~~2~~

$\phi P_1 = 10$
$\phi P_2 = 4+3+2+4 = 13$
$\phi P_3 = 2+1+2+2 = 7$

additivity $= \phi P_1 + \phi P_2 + \phi P_3$
$= 10+13+7$
$= 30$ ✓ total {P1 P2 P3}

_orderno- 6678473305 -Sm no 30 to 103

model agnostic is slow compare to other explainer but we can use for any model

- simple data - simple model - good explainability
- complex data need complex model so they works fine but loose interpretability. so thats why its called black box model as if they gives good prediction & high accuracy but have no interpretability. so we need some model which provide or explain the model + why ?
- with explainable AI we can find the answer if model provide certain prediction then why its that so.

SHAP Working -

sharply value - game theory.

- it based on sharply value
- sharply value comes via feature - x
- so we have x and y model output
- shaply value - avg of y or base value.
  - need background data
  - background data mostly be xtrain
- calculate base value
- shap value move base value
  - if its +ve - base value goes to higher
    if its -ve - base " " - lower

SHAP — model Agnostic
model specific

ϕ_0 = avg prediction

$\phi_0 + \sum_{i=1}^{d} \phi_i = \hat{y}$

- shap value have - magnitude — how much original prediction effected
- sign ↘ +ve
-ve

- SHAP uses
↘ mathematical concept - game theory
- calculate individual contribution of each feature for prediction.

LIME - local interpretable model Agnostic explaination

- local explaination only.
- interpretable - explainable
- model agnostic - dont care what model we use.
- Lime train his own model which is simple so interpretable as well.
- ~~SHAP~~ - surrogate model concept.
- implementation feest
- no more complex computations required.

example f1 f2 f3 P
10 20 30 35

so he create perturbation / disturbance.

Actual	f_1	f_2	f_3	$\hat{y}$	
→	10	20	30	135	$\hat{y}$
Random	11	20	28	~~135~~	$\hat{y}_1$
	113	19	81	135	$\hat{y}_2$
	9	20	30	~~135~~	$\hat{y}_3$
	8	18	35		$\hat{y}_4$

slight disturbances —

* like we create syntetic data.

x_{new} y_{new} → interpretable model (example linear)

↓ train

↓ train regression model

↓ called surrogate model

suppose we have linear regression so model trying to find coefficient values

$$\hat{y} = \beta_0 + \beta_1 x_1 \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5$$

as we have 5 feature

once we train the model we get our coef and intercept.

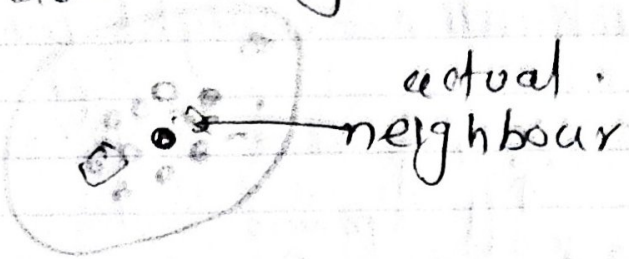
coef — +ve or -ve
which takes prediction in +ve direction
or -ve direction.

—

there is anoter concept - weigh

synthetic data — neighbourhood

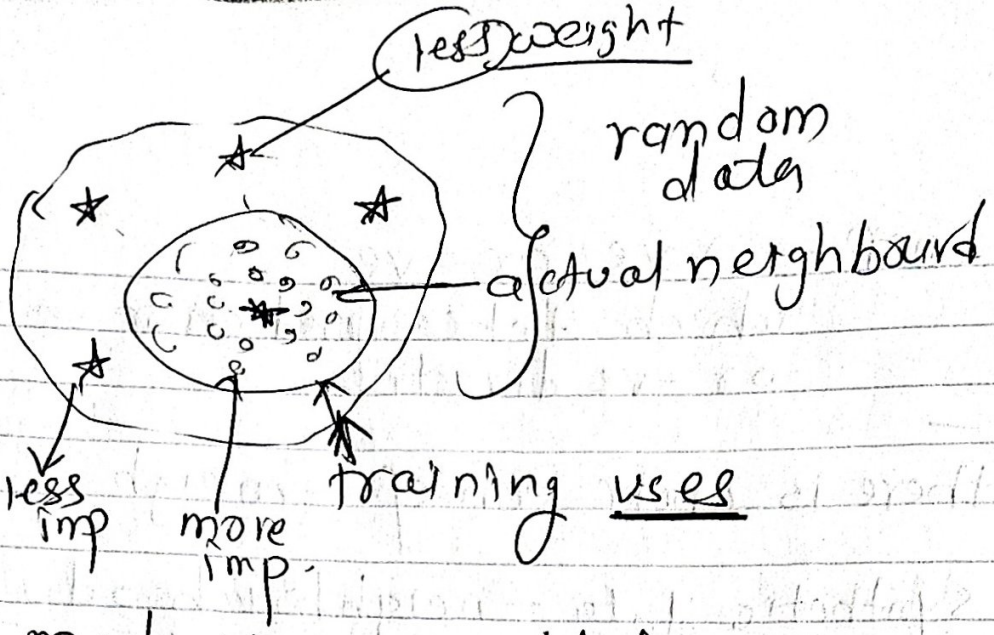
example



why synthetic data — because it shows that the disturbance in the actual value how to affect the prediction.

so here if we create the synthic data or we can say neighbourhood we need the neighbour which dont have big disturbance so those created far from actual point we apply less weight so mode will give not actual learn from them but from actual neighbours

- increase wight for actual neighbourhood
- lesson " " far "
- weight assigned by distances (eludien distances)



weighted machanism – weight base

$$\Pi(x, s) = \exp\left(\frac{-D(x,s)}{\sigma^2}\right)$$

(x: actual, s: syntheric)

we use (kernel) function for weight assignment

Example

pt	x_1	x_2	x_3	x_{new} → f_{new} model pred
1	2·2	4·8	1·1	10·4
2	1·8	5·2	0·9	9·4
3	2·5	5·5	1·0	11·2
4	1·5	4·5	1·5	8·7
5	2·0	6·6	0·5	10·8
6	3·0	5·0	1·0	12·0

↓

Black box model.

new vector	y	distances	weight — with kernel parameter
(2.2, 4.8, 1.1)	10.4	0.300	0.91
(2.0, 6.0, 0.5)	10.8	1.1120	0.2865

(Φ-shap)

$$\hat{y}_{surrogate}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

$$= 3.93172.$$

intercept work as basevalue shap.

if we feed above value to our surrogate mode then we gets the coef - $\beta_0 \beta_1 \beta_2 \beta_3$

so coef values are your explaination

- create synthetic data
- calculate prediction using black box model
- we calculate distances from actual point using eculdren distance
- assign weights → distance less — weight m
  distance more — in les
- so model train on the new data.
  and get intercept and coef value which explain model. easily.

Advantage — fast to implement
- Hardware light

Disadvantage — coef values not stable.

linear mode — syntetic data
random → change train data

3) Lime - no mathamatical approach
shap - have mathamatic grounding eg game theo
4) surrogate - Linear regration

Non linear dataset → not work well.

* Lime - not used in production set up
- as use random cyntetic data every time so not trustworthy.

- local explaination / No Global explanation
- Not more plots in LIME.