ML Week 2 - Supervised I

1. linear Regression
2. Decision trees
3. knneerest Neighbours

} this week

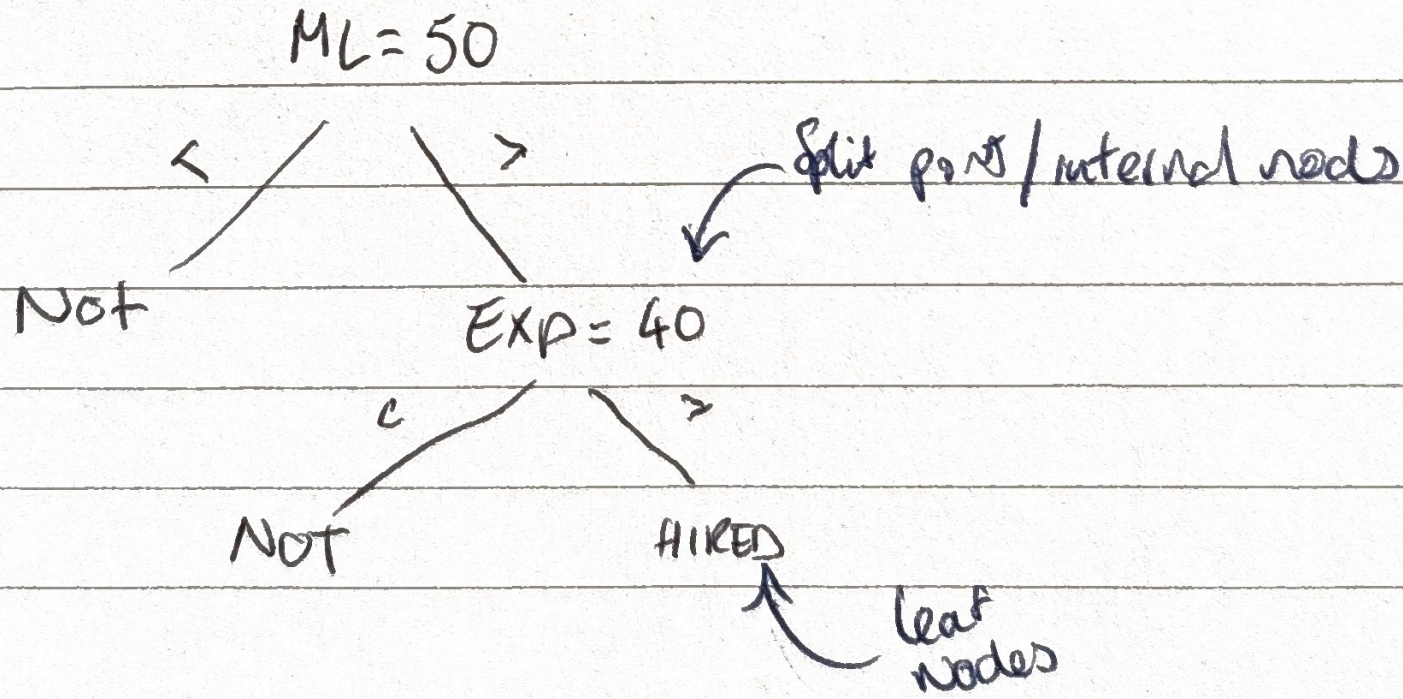
Decsion trees

Recall, the goal of an ML is to find a function $f(\cdot)$ that takes in x_n & outputs y_n.

- $f(x) = \hat{y} = y$ — TR set
- $f(x_m) = \hat{y}_m = y_m$ where $x_m \notin \{x_n\}$

example:

Feature: Exp, ML grade output: Hired



- DTs work by partioning the data recursily — binary tree structure
- Split point defined by a hyperplane that seperates partitions
- leaf nodes = labels for Datas instances that are the same
- Used for Classif + regression

Purity

- Purity helps decide when to split or not

$$\text{Purity}_m = \max_{c_k} \frac{n_{mk}}{n_m}$$ where n_m =

n_m = num data points in the partition region m

n_{mk} = num data points that belong to label class c_k in the partition region m

Max purity = 1

e.g. ML = 51 C_1 = Not C_2 = Hired

L51 = 3 $C_1 = \frac{3}{3}$ $C_2 = \frac{0}{3}$ $\max_{c_k}\{\frac{3}{3}c_1, \frac{0}{3}c_2\}$

$= \frac{3}{3} = 1$ purity, not need to split furthes

A purity of 1 means the whole cohort of a category/split was partitioned

others, >51, max$_k$ $\{\frac{2}{5}, B_1, \frac{3}{5}(2\}$ = 0.67
less than 1 so split further

How to determin the split point?
- optimal point determin by a split criterean
- to types: info gain & gini index

Information Gain: Entrophy
- Entrophy = measure of uncertainty
- Must compute entropy then info gain
- choose split w/ highest info gain

Gini index
- Computing formular
- Compute Gini for both sides of split
- lowest index is choose

Example: best split point

- order the data q a parameter
- Calc split potential point inbetween all existing pounts - min Points instances
- Count split outcomes 0=, 1=

$n_{M0<}$ $n_{M1<}$, $n_{M1>}$, $n_{M0<}$

- Plug into Gini or INFO formula

Gini Example

$$G_{M<S} / M_{>S} = \frac{n_{M<S}}{n_M}\left(1-\sum_{k=1}^{K}\left(\frac{n_{MK<S}}{n_{M<S}}\right)^2\right)+\frac{n_{M>S}}{n_M}\left(1-\sum\left(\frac{n_{MK>S}}{n_{M>S}}\right)^2\right)$$

less than side of split / more than

27.5 Split

$$\frac{1}{8}\left(1-\left(\left(\frac{0}{1}\right)^2+\left(\frac{1}{1}\right)^2\right)\right)+\frac{7}{8}\left(1-\left(\left(\frac{3}{7}\right)^2+\left(\frac{4}{7}\right)^2\right)\right)=0.43$$

8 = Total , 1, 7 = Splits , inner = Split total yes & no

- Calc for all mid pounts & select the best metric

Categorical split points:
- Each ~~point has a~~ value/categ is a potential split point
- Otherwise same process

Building An Optimised Model
- purity threshold
- Max num of leaf Nodes } parameters
- inital region
- Purity of region

region = the training data used

if purity ≥ threshold
stop splitting

Else
- for each feature find best split point
- Select best feature
- Split using feature > split
- Repeat for each new region created

Overfitting w/ tree models:

- = large tree with many nodes
- address overfit by combining nodes

Pruning

- reduce overfit by removing nodes
- Prune criterea used to det if given node should be pruned

$$V_T = \sum^{|T|} \mathcal{E}_T + \alpha|T|$$

$|T|$ = total leaf nodes in Tree

α = Prune hyperpara

$\mathcal{E}_T$ = error measure for T (gini or entropy)

$\mathcal{E}$ = contribution impact

$|T|$ = size factor

Ensemble learning

- Bagging - Bootstrap aggregating
  Bootstrap = each model trained on random subset
  Aggregate = average pred of Sub models

Pros - Capture complexity of data, average out errors, Robust to overfitting as sub models dont see full data

Boosting
- Cumlative training, Multiple Models built from a base model
- Data weighting - worse predicted data is weighted for closer pred on next model
- Prediction weighting - Pred of Community model weighted over the multiple

K-Nearest Neighbours

With KNN you use the classes of the K nearest neighbours of x_j to determine its output $\hat{y}_j$

knn does not produce an model $f(\cdot)$

Step 1: Decide distance metric
Euclidean distance
$dist(x_i, x_j)\ \forall i = 1,2 \ldots N$
for each value of x find its K nearest neighbour based on this metric
Manhattan Distance
2 Dimension measure
cosine Similarity

Step 2 Determine $\hat{y}$ of x_i based on the labels of neighbours
Voting / average system

- For class = majority regression = Average
- need to chose num of neighbours
$\hat{y}_i = \frac{1}{K} \sum y_i$
- can distance-weight to prioritise closer points

time Complexity for knn

$O(N^2D)$ N = data instances
D = total number feature

KNN is computationally heavy / many points, lots of calculations

each data point M is compared to each of N data points & each data point has D numb of dimensions to be compared

$Dist(x_1, x_2) = \sqrt{\sum_{d=1}^{D} (x_{1d} - x_{2d})^2}$, $\forall i, i = 1, 2, \ldots N$, $i \neq 1$, $\forall j$, $j = 1, 2, 3 \ldots$ M

- Euclidean

Minimizing Complexity

- indexing
- kd tree = ~~id~~ indexing tree
- recursively partion the data region
- Similar to decision tree

- Reduce Dimensions ← week 6