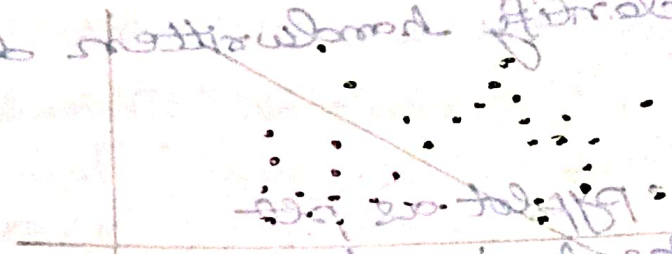
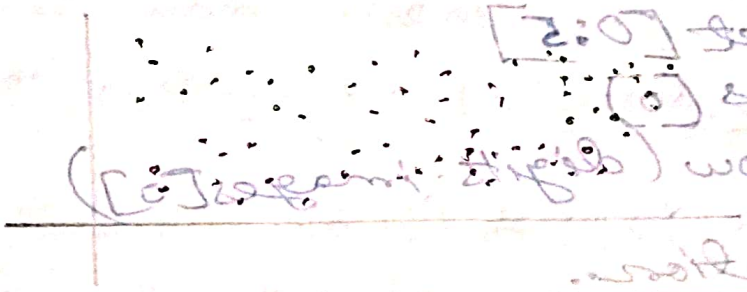
Decision Tree Algorithm..

When data is like this you can easily draw a decision boundary as a straight line and you can use linear regression to do the prediction.



When data is complex.. It is difficult to find the decision boundary.. ie its not possible to draw a sigle line to create the boundaries. so linear regression is not possible here.



What is a 'Decision Tree

⊙ What is a Decision Tree? How does it work

↳ Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems.

↳ It works for both categorical and continuous input and output variables.

↳ In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variable

⊙ Categorical Variable Decision Tree: Decision Tree which has categorical target variable then it called as categorical variable decision tree. Example:- In above ~~sentence~~ scenario of student problems where the target variable was "Student will play cricket or not" i.e. YES or NO.

⊙ Continuous Variable Decision Tree: Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

⊙ Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.

⊙ Splitting: It is a process of dividing a node into two or more sub-nodes.

⊙ Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.

⊙ Leaf/Terminal Node: Nodes do not split is called Leaf or Terminal node.

⊙ Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.

⊙ Branch / Sub-Tree - A sub section of entire tree is called branch or sub-tree.

⊙ Parent and Child Node: A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

Classification by Decision Tree Induction

⊙ Decision tree
- ↳ A flow-chart-like tree structure.
- ↳ Internal node denotes a test on an attribute.
- ↳ Branch represents an outcome of the test.
- ↳ Leaf nodes represent class labels or class distribution.

⊙ Decision tree generation consists of two phases

⊙ Tree construction.
- At start, all the training examples are at the root.
- Partition examples recursively based on selected attributes.

⊙ Tree pruning.
- Identify and remove branches that reflect noise or outliers

⊙ Use of decision tree classifying an unknown sample.
- Test the attribute values of the sample against the decision tree.

Algorithm for Decision Tree Induction

⊙ Basic algorithm (a greedy algorithm)
- ↳ Tree is constructed in a top-down recursive divide-and-conquer manner
- ↳ At start, all the training examples are at the root.
- ↳ Attributes are categorical (if continuous-valued, they are discretized in advance)

↳ Examples are partitioned recursively based on selected attributes.

↳ Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)

• Conditions for stopping partitioning

↳ All samples for a giving node belong to the same class.

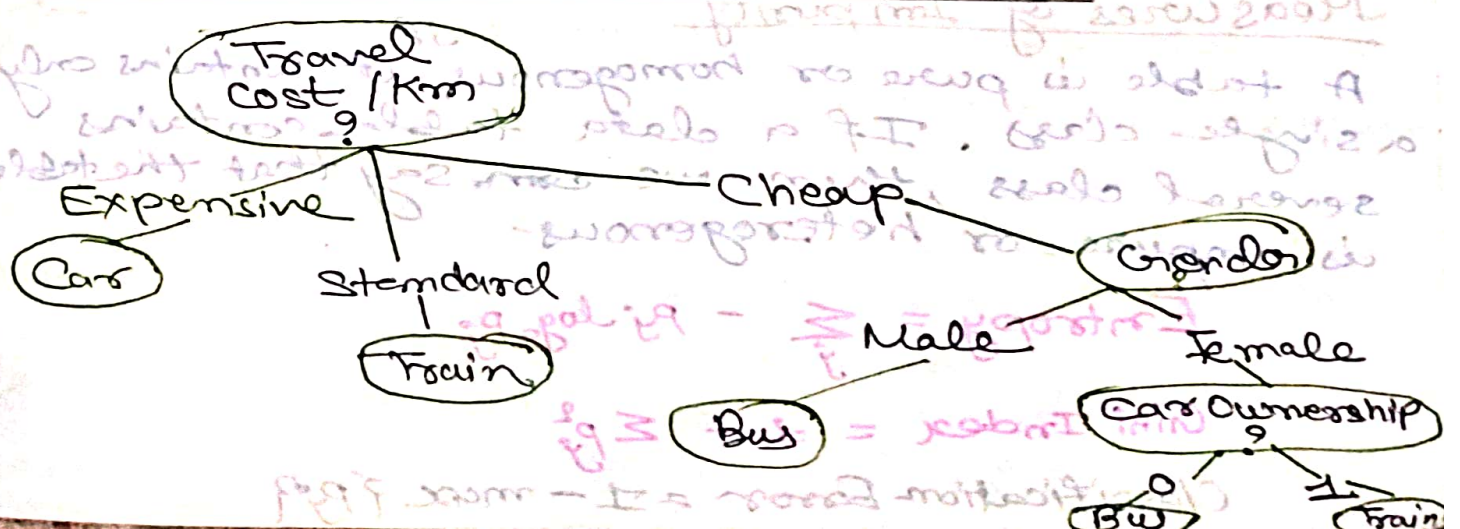
↳ There are no remaining attributes for further partitioning - majority voting is employed for classifying the leaf.

↳ There are no samples left.

Example I / problem I

	Attributes			Classes
Gender	Car ownership	Travel cost ($)/ Km	Income Level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Decision Tree for example 1



Rules Generated

Rule1

If travel cost/km is expensive then
mode = car

Rule2

If Travel cost/km is standard then
mode = train.

Rule3

If Travel cost/km is cheap and gender is male then, mode = bus

Rule4

If Travel cost/km is cheap and gender is female and she owns no car then,
mode = bus

Rule5 :

If Travel cost/km is cheap and gender is female and she owns 1 car then,
mode = train.

Classification using Decision Tree

Person name	Gender	Car Ownership	Travel cost ($)/km	Income Level	Transport Mode
Alex	Male	1	Standard	High	Train
Buddy	Male	0	Cheap	Medium	Bus
Cherry	Female	1	Cheap	High	Train

Measures of Impurity

A table is pure or homogenous, if it contains only a single class. If a data table contains several class, then we say that the table is impure or heterogenous.

$$\text{Entropy} = \sum_j - p_j \log_2 p_j$$

$$\text{Gini Index} = 1 - \sum p_j^2$$

$$\text{Classification Error} = 1 - \max[p_j]$$

Probability of classes

$$Prob(Bus) = \frac{4}{10} = 0.4$$

$$Prob(Car) = \frac{3}{10} = 0.3$$

$$Prob(Train) = \frac{3}{10} = 0.3$$

Entropy

$$Entropy = \sum_j -p_j \log_2 p_j$$

$$Entropy = -0.4 \log_2(0.4) - 0.3 \log_2(0.3) - 0.3 \log_2(0.3)$$

$$= 0.5288 + 0.5211 + 0.5211 = 1.571$$

The logarithm is base 2: $\log_2 a = \frac{\log_{10} a}{\log_{10} 2}$

Gini Index

$$Gini\ Index = 1 - \sum_j p_j^2$$

$$Gini\ Index = 1 - (0.4^2 + 0.3^2 + 0.3^2) = 0.660$$

Classification Error

$$Classification\ Error = 1 - \max\{p_j\}$$

$$Classification\ Error\ Index = 1 - Max\{0.4, 0.3, 0.3\}$$

$$= 1 - 0.4$$

$$= 0.60$$

In notebook

```
import pandas as pd
from sklearn import tree
from sklearn.preprocessing import LabelEncoder
```

Label Encoding

Label Encoding refers to converting the labels into a numeric form so as to convert them the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

Load the data..

```
data = pd.read_csv('salaries.csv')
inputs = data.drop(['salary_more_then_100K'], axis='columns')
target = data['salary_more_then_100K']
```

Data preprocessing

```
le_data = LabelEncoder()
company_l = le_data.fit_transform(inputs['company'])
job_l = le_data.fit_transform(inputs['job'])
degree_l = le_data.fit_transform(inputs['degree'])
inputs['company'] = company_l
```

Create model predict..

```
model = tree.DecisionTreeClassifier()
model.fit(inputs, target)
model.score(inputs, target)
model.predict([[2,2,0]])
```

data.

Assignment..

In this file using following columns build a model to predict if person would survive or not.

① Pclass ② Sex ③ Age ④ fare.