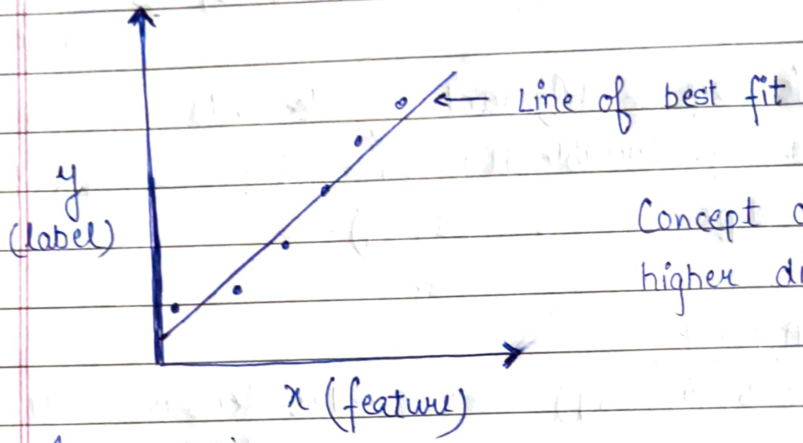
Avishrant

→ Evaluating Tensors
with ~~tf~~ tf. Session () as sess :
tensorname . eval ()

—— x —— x —— x —— x —— x —— x ——

Tensorflow Algos

Major
* Linear Regression (Line of best fit). It is of the form
$y = mx+b$



Concept can be extended to higher dimensions.

→ Implementation
Training - Feeding of data points to an ML model so that it can infer the rule that guides the whole data set.

Epochs = The number of passes of the entire dataset the ML algorithm has completed. We feed the dataset or batches to our model again and again in hope that the model will better determine how to estima it.

Input Function (Tensorflow) = Convert pandas dataframe to tf. data. Dataset objects with appropriate batch-size and epochs.

For batch-size and epochs :
data. batch(size). repeat (epoch-number)

After using the input function for creation of training and testing dataset, it's time for training the model.

- For Linear classifier,

  est = tf. estimator. LinearClassifier (feature-columns = fcs)

* feature-column represents what values are actually present in a particular feature. Can be both string and numbers.

Training = est. train (training - input - function)

Evaluation = est. evaluate (eval - input - function)

It returns a dictionary which contains information about the evaluation such as accuracy, mean, auc etc.

Prediction = result = est. predict (dpoint - input - func)

Steps :

1. Imports (Obvious)
2. Load Datasets (Input function (epoch, batch ...))
3. Get feature columns
4. Get your model ready (DNNClassifier, LinearClassifier)
5. Train the model
6. Evaluate and Predict
7. Enjoy

Clustering - It involves grouping of data into discrete sets. Ideally data points that are in the same group should have similar properties. Procedure = (K-Means)

- Randomly pick k points to place k centroids
- Assign data points to the centroid by distance.
- Place the centroids at the center of mass of the correspondi clusters.

- Reassign every point once again to closest centroid.
- Repeat until no point changes its cluster.

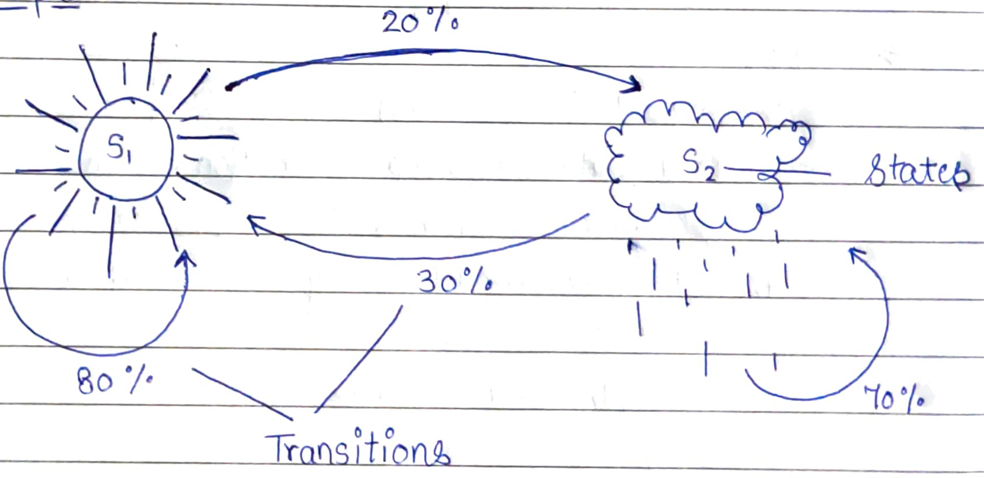
→ <u>Hidden Markov Model</u>

A hidden markov model is a finite set of states, each of which is associated with a probability distribution. Transitions among the states are governed by <u>Transition Properties / Probabilities.</u> This model basically works with probabilities to predict future events or states.

A hidden markov model has :-

1. States → Eg :- hot or cold, high or low, r g or b.
2. Observations → Observation of an event based on state.
3. Transitions → The probability of transition between various states.

<u>Example</u> :-



Observation	Mean : 20		Mean - 5
	Min : 15		Min - 5
	Max : 25		Max - 15