**Section A**

1. Define Machine learning with its objectives

The primary purpose of machine learning is to discover patterns in the user data and then make predictions based on these and intricate patterns for answering business questions and solving business problems.

The goal of [machine learning](https://www.teradata.com/Blogs/Wait-Machine-Learning-and-Artificial-Intelligence-Are-Not-the-Same) is often — though not always — to train a model on historical, labelled data (i.e., data for which the outcome is known) in order to predict the value of some quantity on the basis of a new data item for which the target value or classification is unknown.

1. Define goal of the support vector machine (SVM).

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen.

1. List out different algorithms can be classified under Association Rule Learning Algorithms?

Association rule learning is a type of unsupervised learning technique that checks for the dependency of one data item on another data item and maps accordingly so that it can be more profitable. It tries to find some interesting relations or associations among the variables of dataset. It is based on different rules to discover the interesting relations between variables in the database.

The association rule learning is one of the very important concepts of [machine learning](https://www.javatpoint.com/machine-learning), and it is employed in **Market Basket analysis, Web usage mining, continuous production, etc.**

Association rule learning can be divided into three types of algorithms: 1. Apriori 2. Eclat 3. F-P Growth Algorithm

1. Define any algorithm you know in to solve a problem in Reinforcement Learning.

There are two important learning models in reinforcement learning:

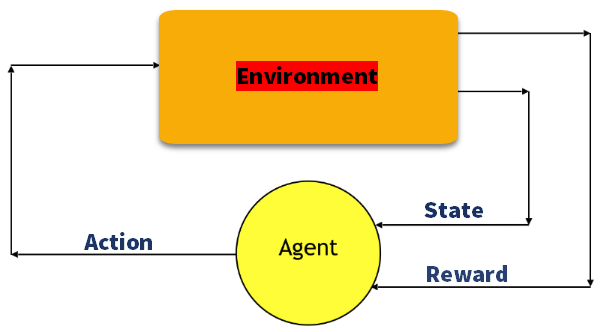
* Markov Decision Process
* Q learning

**Markov Decision Process**

The following parameters are used to get a solution:

* Set of actions- A
* Set of states -S
* Reward- R
* Policy- n
* Value- V

The mathematical approach for mapping a solution in reinforcement Learning is recon as a Markov Decision Process or (MDP).

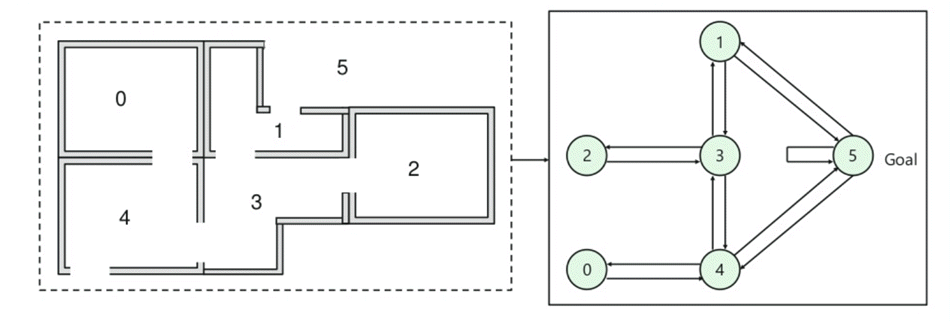


**Q-Learning**

Q learning is a value-based method of supplying information to inform which action an agent should take.

Let’s understand this method by the following example:

* There are five rooms in a building which are connected by doors.
* Each room is numbered 0 to 4
* The outside of the building can be one big outside area (5)
* Doors number 1 and 4 lead into the building from room 5



Next, you need to associate a reward value to each door:

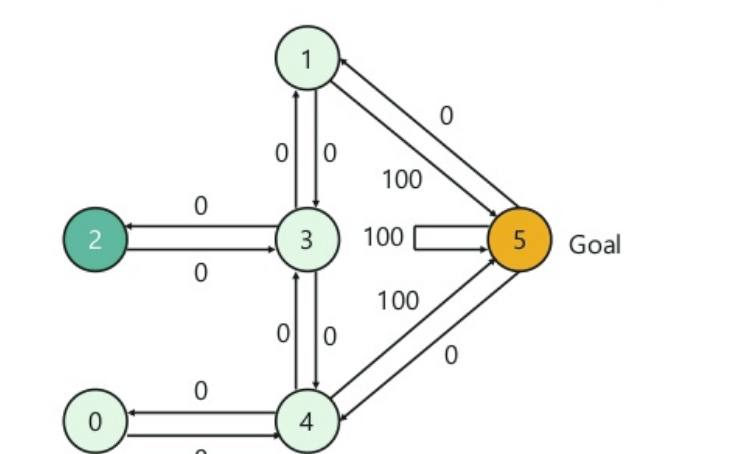
* Doors which lead directly to the goal have a reward of 100
* Doors which is not directly connected to the target room gives zero reward
* As doors are two-way, and two arrows are assigned for each room
* Every arrow in the above image contains an instant reward value

**Explanation:**

In this image, you can view that room represents a state

Agent’s movement from one room to another represents an action

In the below-given image, a state is described as a node, while the arrows show the action.



For example, an agent traverse from room number 2 to 5

* Initial state = state 2
* State 2-> state 3
* State 3 -> state (2,1,4)
* State 4-> state (0,5,3)
* State 1-> state (5,3)
* State 0-> state 4

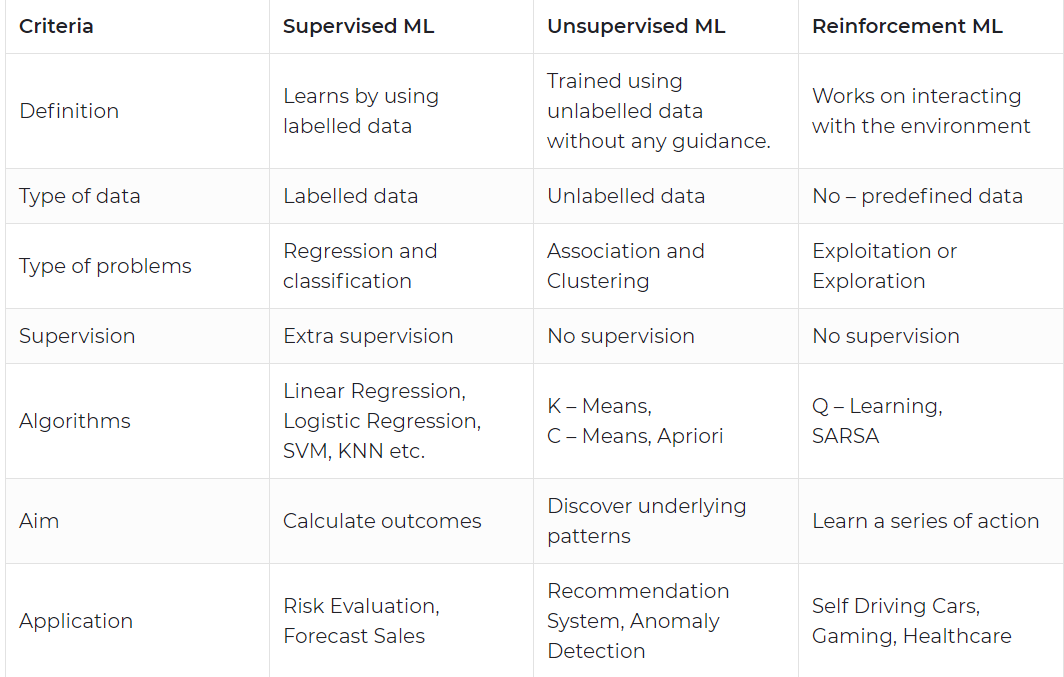
1. Define Agglomerative Clustering and divisive clustering.

**Agglomerative Clustering:**Also known as bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering. This clustering algorithm does not require us to prespecify the number of clusters. Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

**Divisive clustering:** Also known as a top-down approach. This algorithm also does not require to prespecify the number of clusters. Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been split into singleton clusters.

**Section B**

1. Differentiate between Supervised, Unsupervised and Reinforcement Learning



1. How would you detect overfitting in Linear Models?

**Overfitting**is a modeling error that occurs when a function or model is too closely fit the training set and getting a drastic difference of fitting in test set. **Overfitting** the model generally takes the form of making an overly complex model to explain Model behavior in the data under study.

So the first step to finding the Overfitting is to split the data into the Training and Testing set. **If our model does much better on the training set than on the test set, then we’re likely overfitting.** The performance can be measured using the percentage of accuracy observed in both data sets to conclude on the presence of **overfitting**. If the model performs better on the training set than on the test set, it means that the model is likely **overfitting**.

In linear regression overfitting occurs when the model is "too complex". This usually happens when there are a large number of parameters compared to the number of observations.

8. Elaborate relationship between k-Means Clustering and PCA?

**Principal Component Analysis (PCA)**is a tool for dimension reduction. This technique is to transform the larger dataset into a smaller dataset by identifying the correlations and patterns with preserving most of the valuable information.

This is need for feature selection of a model. PCA aims to capture valuable information explaining high variance which results in providing the best accuracy.

**K-Means Clustering**:

It is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups(clusters) where each data point belongs to only one group.

K-means clustering uses “centroids”, K different randomly-initiated points in the data, and assigns every data point to the nearest centroid. After every point has been assigned, the centroid is moved to the average of all of the points assigned to it.

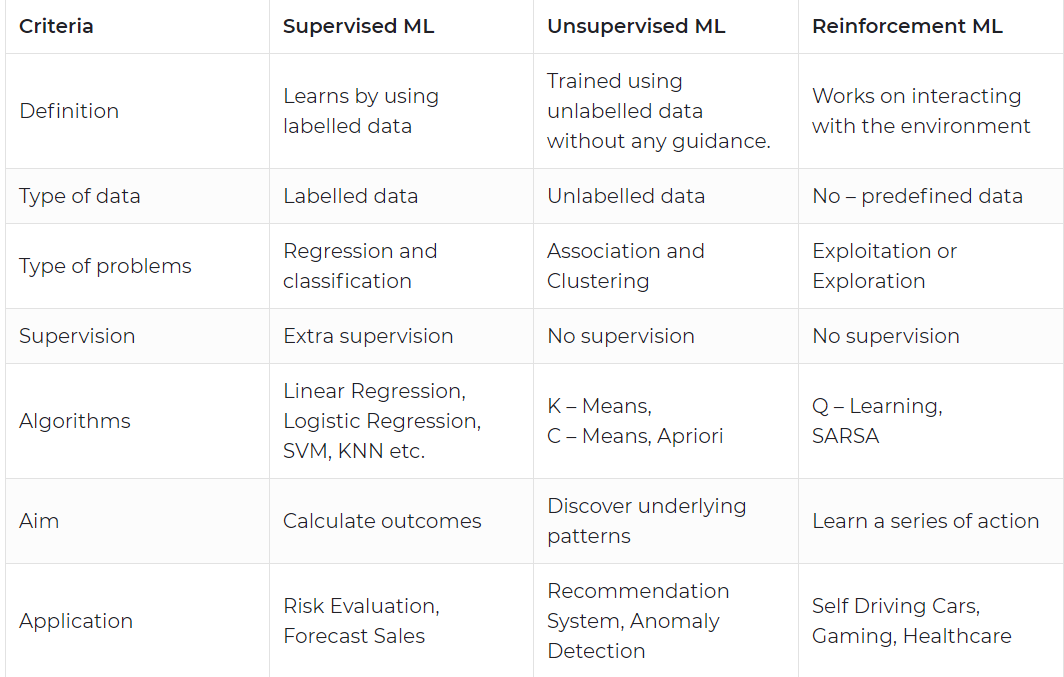
**Relationship between PCA and K Means Clustering:**

Both uses are in dimensionality reduction for visualizing patterns in data from parameters(variables).PCA in conjunction with K-means is a powerful method for visualizing high dimensional data.

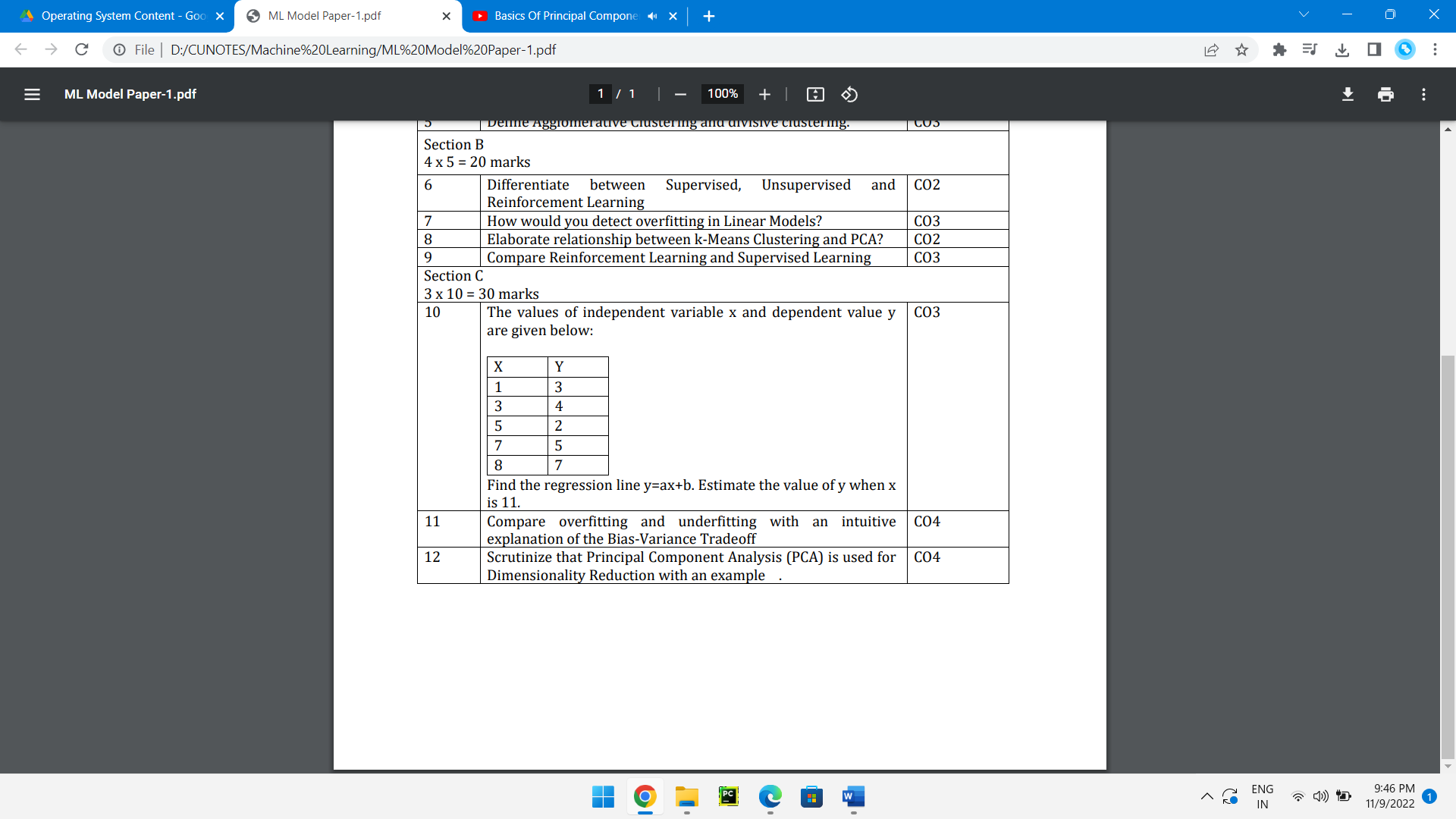
k-means tries to find the least-squares partition of the data. PCA finds the least-squares cluster membership vector. The first Eigenvector has the largest variance, therefore splitting on this vector (which resembles cluster membership, not input data coordinates!) means maximizing between cluster variance.

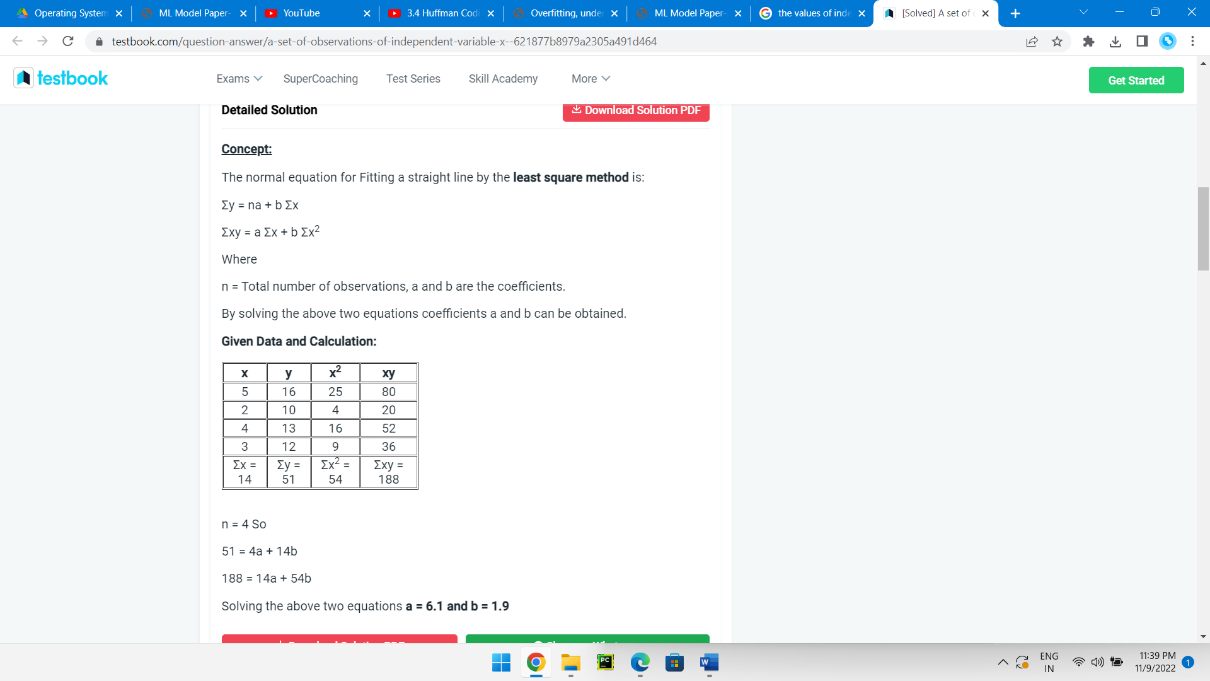
However, when we employ PCA prior to using K-means we can visually separate almost the entire data set. That was one of the biggest goals of PCA - to reduce the number of variables by combining them into bigger, more meaningful features. Not only that, but they are 'orthogonal' to each other.

9. Compare Reinforcement Learning and Supervised Learning.



10.





11. Compare overfitting and underfitting with an intuitive explanation of the Bias-Variance Tradeoff.

Overfitting, underfitting, and the bias-variance tradeoff are foundational concepts in machine learning. A model is **overfit** if performance on the training data, used to fit the model, is substantially better than performance on a test set, held out from the model training process. For example, the prediction error of the training data may be noticeably smaller than that of the testing data. Comparing model performance metrics between these two data sets is one of the main reasons that data are split for training and testing. This way, the model’s capability for predictions with new, unseen data can be assessed.

When a model overfits the training data, it is said to have **high variance**. One way to think about this is that whatever variability exists in the training data, the model has “learned” this very well. In fact, too well. A model with high variance is likely to have learned the noise in the training set. Noise consists of the random fluctuations, or offsets from true values, in the features (independent variables) and response (dependent variable) of the data. Noise can obscure the true relationship between features and the response variable. Virtually all real-world data are noisy.

If there is random noise in the training set, then there is probably also random noise in the testing set. However, the specific values of the random fluctuations will be different than those of the training set, because after all, the noise is random. The model cannot anticipate the fluctuations in the new, unseen data of the testing set. This why testing performance of an overfit model is lower than training performance.

Overfitting is more likely in the following circumstances:

* There are a large number of features available, relative to the number of samples (observations). The more features there are, the greater the chance of discovering a spurious relationship between the features and the response.
* A complex model is used, such as deep decision trees, or neural networks. Models like these effectively engineer their own features, and have an opportunity develop more complex hypotheses about the relationship between features and the response, making overfitting more likely.

At the opposite end of the spectrum, if a model is not fitting the training data very well, this is known as **underfitting**, and the model is said to have **high bias**. In this case, the model may not be complex enough, in terms of the features or the type of model being used.

12. Scrutinize that Principal Component Analysis (PCA) is used for Dimensionality Reduction with an example.

Principal Component Analysis(PCA) is one of the most popular linear dimension reduction algorithms. It is a projection based method that transforms the data by projecting it onto a set of orthogonal(perpendicular) axes.

Reducing the number of input variables for a predictive model is referred to as dimensionality reduction.

Fewer input variables can result in a simpler predictive model that may have better performance when making predictions on new data.

Perhaps the most popular technique for dimensionality reduction in machine learning is [Principal Component Analysis](https://machinelearningmastery.com/calculate-principal-component-analysis-scratch-python/), or PCA for short. This is a technique that comes from the field of linear algebra and can be used as a data preparation technique to create a projection of a dataset prior to fitting a model.

With all the effectiveness PCA provides, but if the number of variables is large, it becomes hard to interpret the principal components. PCA is most suitable when variables have a linear relationship among them. Also, PCA is susceptible to big outliers.

There are many methods for Dimensionality Reduction like PCA, ICA, t-SNE, etc., we shall see PCA (Principal Component Analysis).

Let’s first understand what is information in data. Consider the following imaginary data, which has Age, Weight, and Height of people.

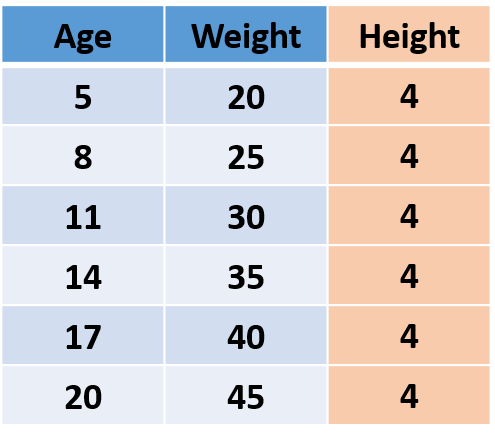
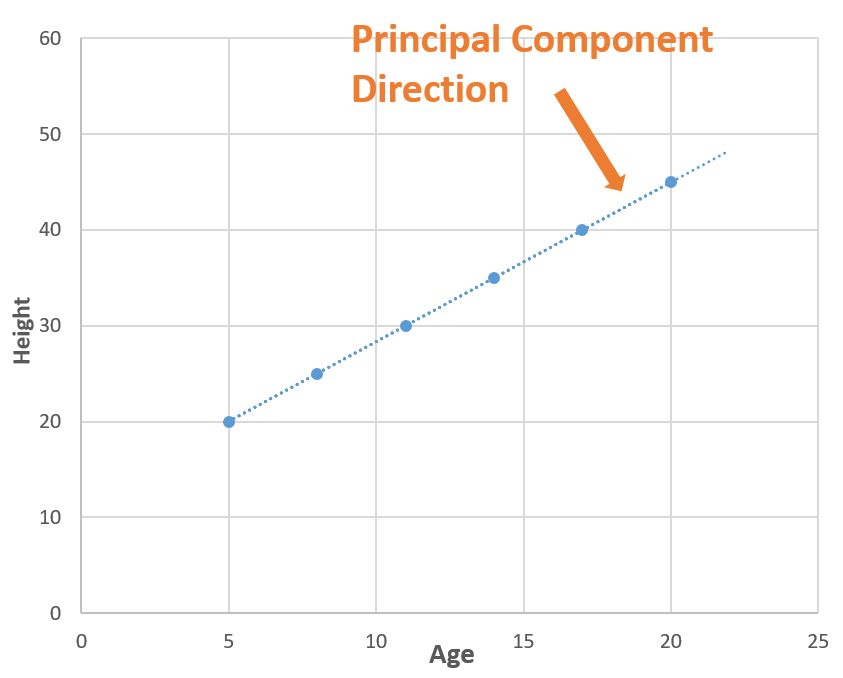


Fig.1

## **The**information lies in the variance!

As the ‘Height’ of all the people is the same i.e. the variance is 0 thus it’s not adding any information, so we can remove the ‘Height’ column without losing any information.

Now we know that information is variance, let’s understand the working of PCA. In PCA, we find new dimensions (features) that capture maximum variance (i.e. information). To understand this we shall use the previous example. After removing ‘Height’, we are left with ‘Age’ and ‘Weight’. These two features are caring all the information. In other words, we can say that we require 2 features (Age and Height) to hold the information, and if we can find a new feature that alone can hold all the information, we can replace origination 2 features with a new single feature, achieving dimensionality reduction!



Now consider a line (blue dotted line) that is passing through all the points. The blue dotted line is capturing all the information so, we can replace ‘Age’ and ‘Weight’ (2 Dimensions) with the blue dotted line (1 Dimension) without losing any information, and in this way, we have done dimensionality reduction (2 dimensions to 1 dimension). The blue dotted line is called **Principal Component.**