**Bank Customer Satisfaction Prediction Using CNN and Feature Selection**

Feature Selection and CNN

Problem statement:-

In this project we are going to build a neural network to predict if a particular bank customer is satisfies or not. To do this we are going to use Convolutional Neural Networks. The dataset which we are going to use contains 370 features. We are going to use feature selection to select the most relevant features and reduce the complexity of our model.

Dataset

The dataset is an anonymized dataset which contains a large number of numeric variables. The TARGET column is the variable to predict. It equals one for unsatisfied customers and 0 for satisfied customers.

Building the CNN

A Sequential() model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.

Conv1D() is a 1D Convolution Layer, this layer is very effective for deriving features from a fixed-length segment of the overall dataset, where it is not so important where the feature is located in the segment. In the first Conv1D() layer, we are learning a total of 36 filters with size of the convolutional window as 3. The input\_shape specifies the shape of the input. It is a necessary parameter for the first layer in any neural network. We will be using the ReLu activation function. The rectified linear activation function or ReLU  for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.



BatchNormalization() allows each layer of a network to learn by itself a little bit more independently of other layers. To increase the stability of a neural network, batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. It applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.

MaxPool1D() downsamples the input representation by taking the maximum value over the window defined by pool\_size which is 2 in case of the first Max Pool layer of this neural network.

Dropout() is used to randomly set the outgoing edges of hidden units to 0 at each update of the training phase. The value passed in dropout specifies the probability at which outputs of the layer are dropped out.

Flatten() is used to convert the data into a 1-dimensional array for inputting it to the next layer.

Dense() is the regular deeply connected neural network layer. The output layer is a dense layer with 1 neuron because we are predicting a single value. Sigmoid function is used because it exists between (0 to 1) and this facilitates us to predict a binary input.