We use keras and tensorflow, deep learning libraries( used to create end to end deep learning projects)

Keras is like a wrapper on top of tensorflow( can create ann ,rnn, lstm rnn using these)

So if we use tf for writing an ann etc, it requires more code,

While keras provides apis with efficient and less lines of code.

# Problem set: churn\_modelling.csv ---

We are using ann ,which can solve both regression and classification problems.

So based on certain bank data, we will be predicting if the person will leave the bank or not.

1.Basic feature engineering

1. Convert categorical variables into numerical
2. Standardization process

2.We will be doing dropout( like removing some neurons from the hidden layer, so that the model does not overfit.

3. for final, we will calculate loss, use some optimizers🡪 use keras , tensorflow for the same.

4. Once the model is created, we will create pickle or .h5 file format.

5. Later we will use streamlit to create a web app -🡪 in web app, we will integrate all our models and deploy in streamlit cloud.

## We do the project using TensorFlow , as keras is integrated within tensorflow..

Important steps:

1. Create label encoder for binary category : M/F
2. Create one hot encoding for geography..

----Sequential network

---Dense class : used for creating neurons within hidden layer( 64 etc)

--- activation fn : sigmoid, tanh , relu, leaky relu ( gets applied on every neuron)

--- optimizers : useful in back propagation( responsible for updating the weights)

--- loss function:-- reduce

--- metrics :[ accuracy- etc

--- training info-🡪 logs -🡪 tensorboard --- visualization.

Q. How many input neurons to be considered in Sequential ANN model 🡪 that will be equal to X\_train.shape() ..that is no of columns

## also when I am creating my dense layer , will be specifying my neurons for hidden layer, but will also tell how many input neurons will be connected to one hidden layer neuron,, that will be --- X\_train.shape[1],)

## moving ahead when we define our second hidden layer , no need to provide input\_shape as it knows that now it will be hidden layer prior to it.

## early stopping : if the loss value is not decreasing , we can apply early stopping , like it doesn’t have to run 5000 epocs, if loss doesn’t decrease , the training can stop there itself

* Patience is till how many epocs it will see the loss
* Restrore the best weight, at what epocs did you find the best weight( in fwd and bck prog) –consider that when you stop the early stopping