**Making the Data Balanced without any Bias to avoid Overfitting and making wrong predictions.**

Clean data plays a huge role in how our machine learns and performs prediction. Here we check the accuracy of data with an imbalanced dataset and then do operations to ensure the dataset is balanced using various techniques. Then train the model on the balanced dataset. The results show that balanced datasets help improve accuracy of model and perform reliable predictions.

We will use the customer churn prediction dataset. We have already done this project in our previous blogs where the accuracy was around 80%. That was the case when the dataset was imbalanced, here we balance the dataset using various techniques and check how it affects the accuracy, is it helpful or not?

This is a classification task wherein we predict if the customer will stay with us or move to some other organization to use their services. To achieve this, we train the ANN on the training dataset, perform required transformations and go with training. At the end of 10 epochs (going through training data 10 times). Epoch could be termed like a revision, the number of times our model will go through the whole training data for training and adjusting the weights to get the best possible accuracy. We get 81% accuracy on the dataset which is okay.

Techniques to make data balanced:

1. Undersampling:

Suppose you have a dataset which has 400 observations for class 0 and 800 observations for class 1, it is seen that the dataset is imbalanced. To balance the dataset we can remove 400 observations of class 1 and we are left with 400 elements of class 1 and 400 elements of class 0. Hence, we get a balanced dataset. Undersampling because we removed elements from class 1.

1. Oversampling:

Suppose you have a dataset which has 400 observations for class 0 and 800 observations for class 1, it is seen that the dataset is imbalanced. To balance the dataset we can add/ make duplicates 400 observations of class 0 and we get 800 elements of class 1 and 800 elements of class 0. Hence, we get a balanced dataset. Oversampling because we added elements to class 0.

1. SMOTE:

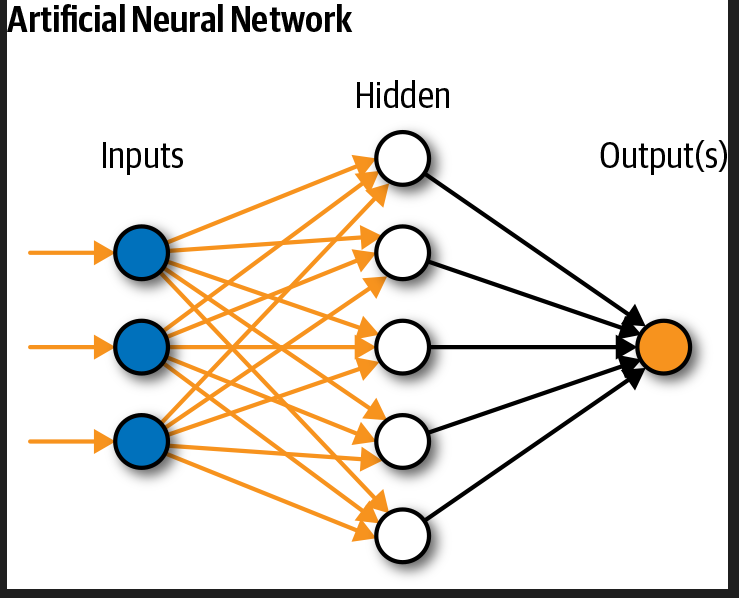
**Synthetic Minority Oversampling Technique, it helps us oversample minority class samples to make the dataset balanced. This is done with help of imblearn library. It works by selecting the points which are related to each other, analyze how they are related and create a new sample based on the analysis**

1. **Ensemble model using undersampling:**

**Ensemble model means training multiple models and adding them up for predicting the output. Models can be trained based on the number of elements in the class with less samples.**

**Artificial Neural Network**

ANN is a neural network which tries to perform tasks like a human does, think like a human brain. Just like a human brain understands things after learning by watching things or by experience, ANN does the same as well. It learns with experience of going through the dataset multiple times and understands the relations, hidden features and parameters. ANN is helpful in doing regression, classification tasks and performs extremely well on huge datasets achieving high accuracy.

****

**Input Layer:**

Whatever input you pass for the model to learn goes through this layer of neural network for performing calculations.

**Hidden Layer:**

The layer as the name suggests hidden because when we see the real time application we only focus on the input and output, we do not focus on how things happen. Hidden layer performs calculations, does processing, understands the hidden features and updates weights to get the best possible accuracy.

**Output Layer:**

**The input passes through hidden layer where processing happens and output is returned.**

**For detailed explanation of ANN, refer to this blog** <https://ai.plainenglish.io/artificial-neural-network-ann-edec89ce4daa> **.**

# How to Execute?

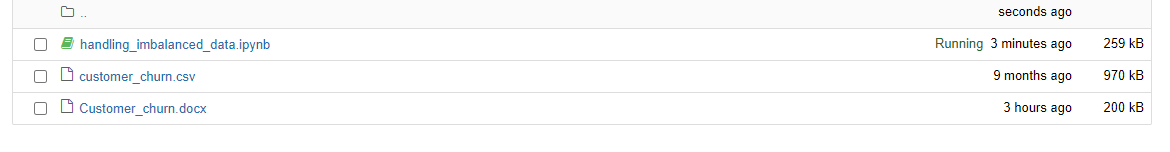
So, before execution we have some pre-requisites that we need to download or install i.e., anaconda environment, python and a code editor. **Anaconda**: Anaconda is like a package of libraries and offers a great deal of information which allows a data engineer to create multiple environments and install required libraries easy and neat.

To check on how to install anaconda, set up jupyter notebook refer to this article. You can skip the article if you have knowledge of installing anaconda, setting up environment and installing requirements.txt

# requirements.txt is included with the zip file provided.

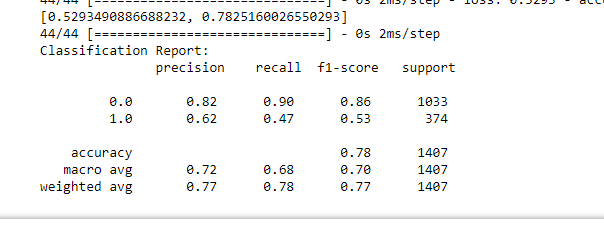
# Steps to execute

To run the code, start jupyter notebook, goto the required file path.



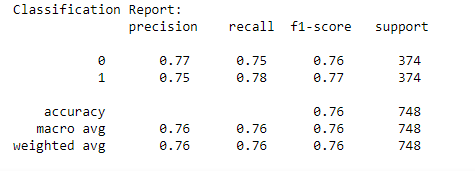
When you run the handling\_imbalanced\_data.ipynb file, you get the appropriate results.

* **Case-1 – Imbalanced dataset**



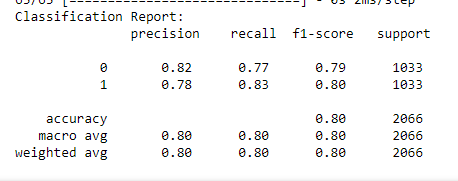
f1-score for 1.0 is 0.53 for imbalanced dataset

* **Case-2 –Balanced dataset using under sampling**

****

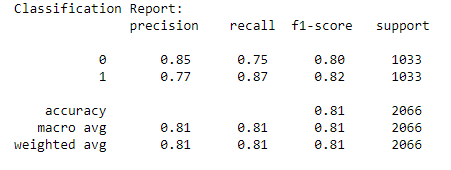
F1-score for class1 increased from 0.53 to 0.77

* **Case-3 –Balanced dataset using over sampling**

****

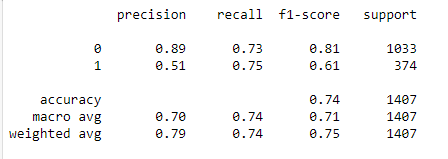
F1-score for class1 increased from 0.53 to 0.80

* **Case-4 –Balanced dataset using SMOTE**



F1-score for class1 increased from 0.53 to 0.81 which is a very good improvement

* **Case-5–Balanced dataset using ensemble with undersampling**



F1-score for class1 increased from 0.53 to 0.61 (not very effective)

# Issues Faced.

* 1. Make sure to change the path in the code. Give full path of dataset/csv file you want to use.
  2. Make sure you have the appropriate versions of tensorflow and keras.