At first

We process the data for classification

You also need to convert all data types for some variables (int, float) to perform the classification process

We did this as shown:

```
def edu_level(x):
    if x=='Graduate'        :   return 0
    if x=='Masters'         :   return 1
    if x=='High School'     :   return 2
    if x=='Phd'             :   return 3
    if x=='Primary School' :   return 4
def major(x):
    if x=='STEM'                    :   return 0
    if x=='Business Degree'         :   return 1
    if x=='Arts'                    :   return 2
    if x=='Humanities'              :   return 3
    if x=='No Major'                :   return 4
    if x=='Other'                   :   return 5
def company_t(x):
    if x=='private limited company':return 0
    if x=='Funded Startup'          :   return 1
    if x=='Early Stage Startup'     :   return 2
    if x=='Other'                   :   return 3
    if x=='Public Sector'           :   return 4
    if x=='Non-Governmental Organisation': return 5

test_clean2['education_level'] = test_clean2['education_level'].apply(edu_level)
test_clean2['major_discipline'] = test_clean2['major_discipline'].apply(major)
test_clean2['company_type'] = test_clean2['company_type'].apply(company_t)
```

- Then we print out information about the DataFrame, including index type, columns, non-blank values, and memory usage, to ensure that the data is ready for the classification process.

```
test_clean2.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8877 entries, 0 to 8876
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   enrollee_id             8877 non-null   int64
 1   city_development_index  8877 non-null   float64
 2   gender                  8877 non-null   int64
 3   relevent_experience     8877 non-null   int64
 4   enrolled_university     8877 non-null   float64
 5   education_level         8877 non-null   int64
 6   major_discipline        8877 non-null   int64
 7   experience              8877 non-null   int64
 8   company_type            8877 non-null   int64
 9   last_new_job            8877 non-null   int64
 10  training_hours          8877 non-null   int64
 11  target                  8877 non-null   float64
 12  company_size_from       8877 non-null   int64
 13  company_size_to         8877 non-null   int64
 14  city                    8877 non-null   int64
dtypes: float64(3), int64(12)
memory usage: 1.0 MB
```

- We selected the k-nearset neighbor method
  - K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Now we will explain the method we mentioned and the accuracy of the classification method used

Model classification

In [780]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
# split data
X = test_clean2.drop("target", axis=1)
y = test_clean2["target"]
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.333)
model =  KNeighborsClassifier()
model.fit(X_train, y_train);
model.score(X_test, y_test)
```

Out[780]: 0.8207642881298614