Week 9 Data Glacier

Team Name: Data 4 Science

Team Members Details:

- ### Amshumann Singh:

  amshumannsingh@gmail.com
  Ireland
  Dublin City University
  Data Science

- ### Hamza Al Hajj Chehade:

  hamza.h.chehade@gmail.com
  Lebanon
  Lebanese University – Faculty of Sciences
  Data Science

- ### Vaishnavi Dixit:

  withmylaptop@gmail.com
  India
  SIES GST, Mumbai University
  Data Science

Problem Description:

ABC bank wants to sell its term deposit product to customers and before launching the product they want to develop a model which help them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution).

The Bank wants to use ML model to shortlist customer whose chances of buying the product is more so that their marketing channel (tele marketing, SMS/email marketing etc.) can focus only to those customers whose chances of buying the product is more. This will save resource and time (which is directly involved in the cost (resource billing)).

GitHub Link:

https://github.com/HamzaAlHajjChehade/Bank_Marketing_Campaign

https://github.com/Vaishnavi-Dixit-12/Bank_Marketing_Campaign/blob/main/week9_dataglacier.ipynb

Data Cleansing and Transformation:

1. We choose the bank_additional_full dataset
2. Checking for null values: There is no NA values in our dataset
3. Checking for duplications: There is duplications and these duplicates does not show that there exist clients with the same details, but it shows that duplicates occur while entering the data. Thus, they must be dropped.
4. Handling Outliers: We can notice that many features have no outliers except for the age, campaign, and duration features. The outliers in the previous features are handled using the Z Score Method such as:

```
#Eliminating Outliers for colx
upper = data[colx].mean() + 3*data[colx].std()
lower = data[colx].mean() - 3*data[colx].std()
print(f'Upper limit for col - colx is {upper} & lower limit is {lower}')
data = data[(data[colx] < upper) & (data[colx] > lower) ]
data.shape
```

5. Data Transformation: Applied label encoder for all the categorical values/columns in the ‘categorical’ list.

```
categorical = [ 'job', 'marital', 'education', 'default', 'housing', 'loan
','contact', 'month', 'day_of_week', 'poutcome','y']
```