* Q1: Build a ML model to identify fraudulent user posting fake reviews on Google Maps
  + Logs table : csv
  + Log of all user activities, - demographics, contributions and history.

# logs.csv

Data\_file\_name = ‘logs.csv’

Import pandas as pd

Logs\_df = pd.read\_csv(data\_file\_name)

print(logs\_df.head())

logs\_df.desc()

Logs\_df[logs\_df[‘country’] == ‘India’ and logs\_df[‘age’] >= 20 and logs\_df[‘age’] <= 30]]

# column: ‘review’ (alphanumeric) -> tokenized

textual\_columns = []

Nontextual\_columns = []

For column in logs\_df.columns:

If column.dytpe == “string”:

textual\_columns.append(column)

Else:

nontextual\_columns.append(column)

From sklearn.preprocessing.vectorizer import tfIdfVectorizer

For column in textual\_columns:

logs\_df[column] = tfIdfVectorizer.fit\_transform(logs\_df[column]])

From sklearn.preprocessing import OneHotEncoder

Categorical\_columns = [] # populated manually

For column in Categorical\_columns:

Logs\_df[column] = OneHotEncoder.fit\_transform(logs\_df[column])

From sklearn.preprocessing import MinMaxScaler

Scaled\_columns = [] # populated manually

For column in scaled\_columns:

Logs\_df[column] = MinMaxScaler.fit\_transform(logs\_df[column])

From keras.layers import Dense, Softmax

From keras.models import Sequential

Model = Sequential([

Dense(n), # n = number of features

Dense(n \* 4, activation = ‘relu’),

Softmax(1)

])

model.train(X\_train, y\_train)

Y\_pred = model.predict(X\_test)

From sklearn.metrics import accuracy, precision, recall

accuracy\_score = accuracy(y\_test, y\_pred)

precision\_score = precision(y\_test, y\_pred)

recall\_score = recall(y\_test, y\_pred)