# Bank Marketing Campaign MVP

**Name: Sarah Faisal Alzighaibi**

**Email: Sarah.alnuzha@gmail.com**

## Overview:

In this project our goal is to predict the number of customers who will subscribe a term of deposit in a bank. which help the marketing team with their next campaigns based on previously provided data. Using ANN, logistic regression and xgboost. The data used data provided by Kaggel include more than 41K rows with 20 features. Trying to fulfil the below needs.

## Data understanding:

The data original data shape was (41188, 21) with no null values.

```
In [5]:  data.columns

Out[5]: Index(['age', 'job', 'marital', 'education', 'default', 'housing', 'loan',
               'contact', 'month', 'day_of_week', 'duration', 'campaign', 'pdays',
               'previous', 'poutcome', 'emp.var.rate', 'cons.price.idx',
               'cons.conf.idx', 'euribor3m', 'nr.employed', 'y'],
              dtype='object')
```

```
In [6]:  data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             41188 non-null  int64
 1   job             41188 non-null  object
 2   marital         41188 non-null  object
 3   education       41188 non-null  object
 4   default         41188 non-null  object
 5   housing         41188 non-null  object
 6   loan            41188 non-null  object
 7   contact         41188 non-null  object
 8   month           41188 non-null  object
 9   day_of_week     41188 non-null  object
 10  duration        41188 non-null  int64
 11  campaign        41188 non-null  int64
 12  pdays           41188 non-null  int64
 13  previous        41188 non-null  int64
 14  poutcome        41188 non-null  object
 15  emp.var.rate    41188 non-null  float64
 16  cons.price.idx  41188 non-null  float64
 17  cons.conf.idx   41188 non-null  float64
 18  euribor3m       41188 non-null  float64
 19  nr.employed     41188 non-null  float64
 20  y               41188 non-null  object
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

```
there is no Null values
```

First I clean the data by deleting duplicated and replacing outlier with mean value

```
In [14]: data.duplicated().sum()
         print("There are " + str(data.duplicated().sum()) + " duplicated rows in the dataframe.")
```

```
There are 12 duplicated rows in the dataframe.
```

```
In [15]: data = data.drop_duplicates()
         print("After pre-cleaning, there are " + str(data.shape[0]) + " rows and " + str(data.shape[1]) + " columns in this dataframe
```

```
After pre-cleaning, there are 41176 rows and 21 columns in this dataframe.
```

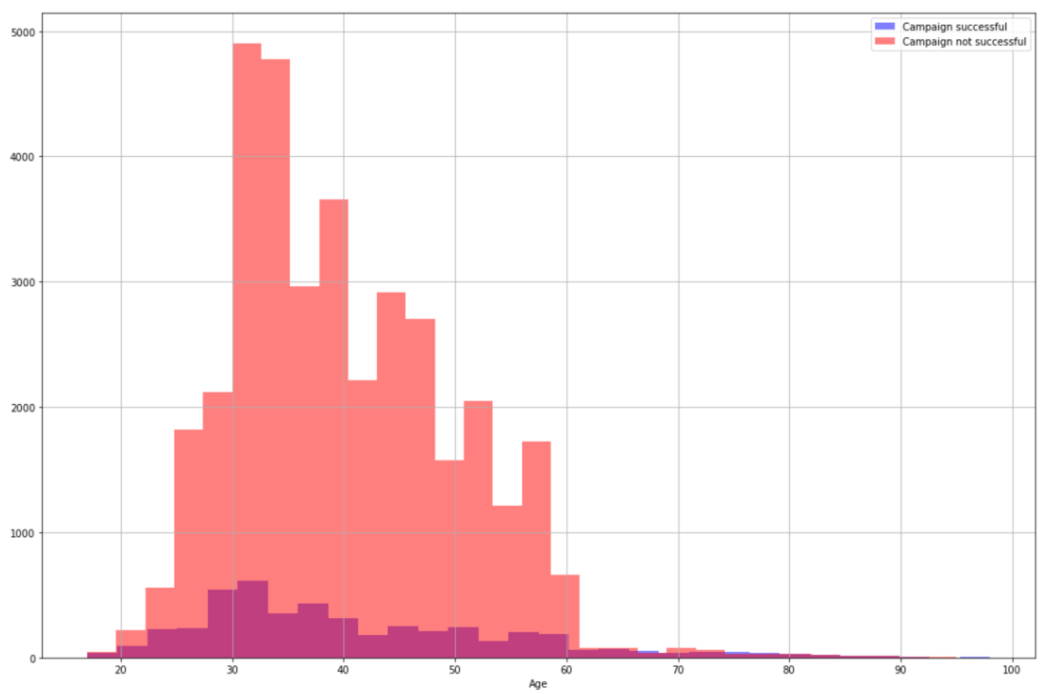
Outlier handling

```
In [25]: outliers = []
         def detect_outliers_iqr(data):
             data = sorted(data)
             q1 = np.percentile(data, 25)
             q3 = np.percentile(data, 75)
             IQR = q3-q1
             lwr_bound = q1-(1.5*IQR)
             upr_bound = q3+(1.5*IQR)
             for i in data:
                 if (i<lwr_bound or i>upr_bound):
                     outliers.append(i)
             return outliers
```

```
In [26]: # Replace outlier values with median

         for col in ['age', 'campaign', 'pdays', 'confidence_index']:
           sample_outliers = detect_outliers_iqr(data[col])
           median = np.median(data[col])
           for i in sample_outliers:
               data[col] = np.where(data[col]==i, median, data[col])
```

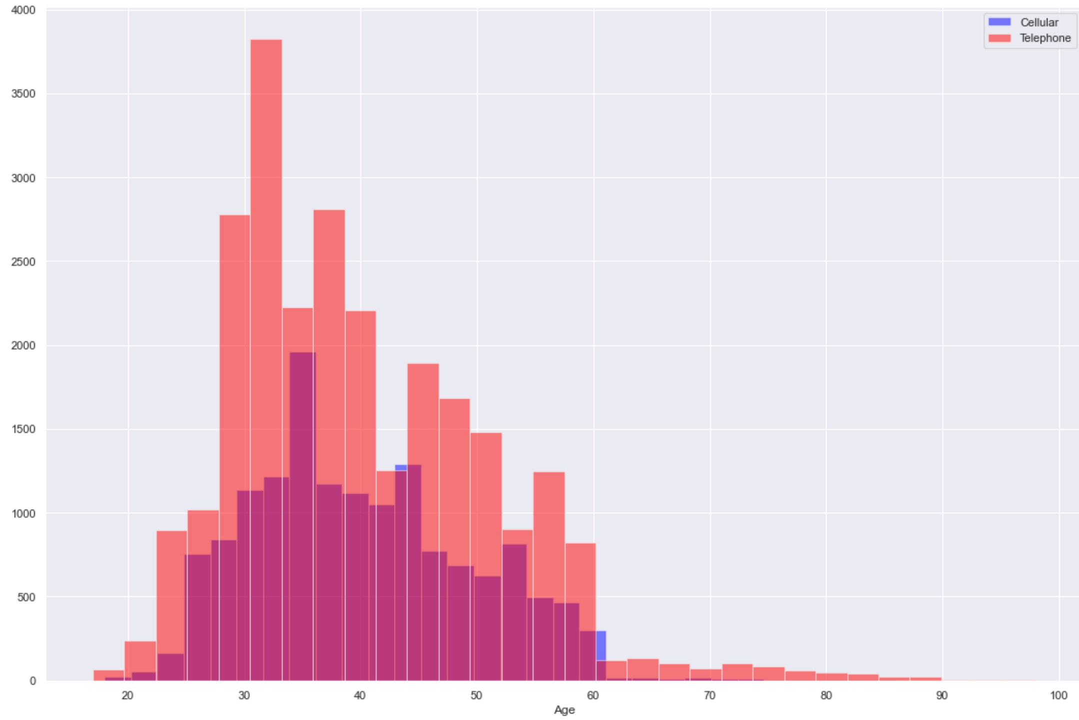
Finding the correlation between variables : (age and successful campaign)




Observation:

Campaign seem to be most successful among younger {< 20 years old} and older clients {>60 years old}

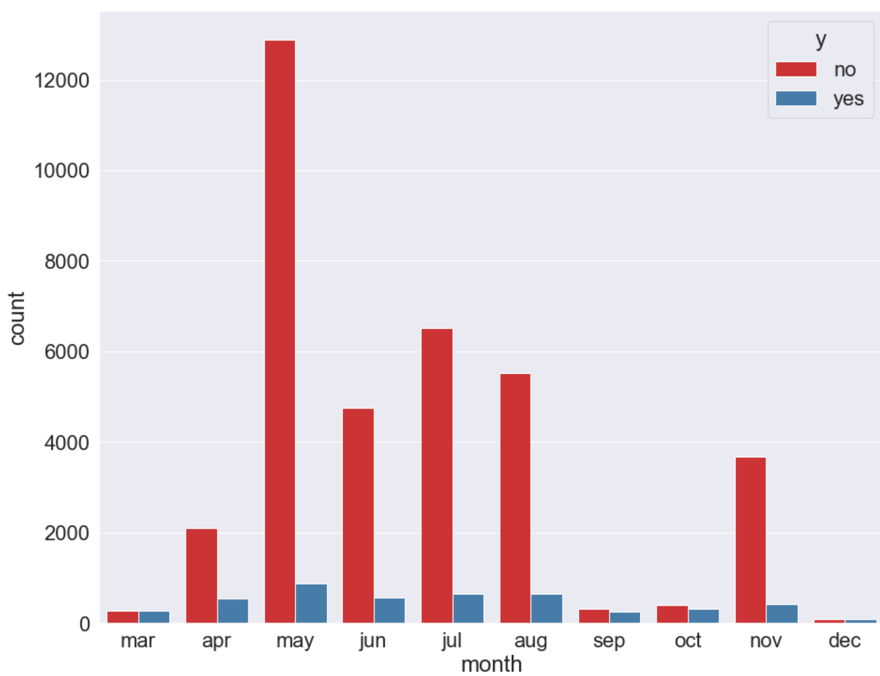
And the way contacting the clints




Observation:

The majority of our clients are contacted through their home phones as opposed to their cellular telephone.

And when is most successful months to do the campaign



# Correlation Matrix:

To find the correlation between features

In [23]:
```
data.corr()
```

Out[23]:

| | age | duration | campaign | pdays | previous | emp_var_rate | price_index | confidence_index | eur_3month | no_of_employees |
|---|---|---|---|---|---|---|---|---|---|---|
| **age** | 1.000000 | -0.000808 | 0.004622 | -0.034381 | 0.024379 | -0.000242 | 0.001009 | 0.129075 | 0.010852 | -0.017607 |
| **duration** | -0.000808 | 1.000000 | -0.071765 | -0.047556 | 0.020600 | -0.027941 | 0.005303 | -0.008126 | -0.032861 | -0.044672 |
| **campaign** | 0.004622 | -0.071765 | 1.000000 | 0.052606 | -0.079182 | 0.150786 | 0.127826 | -0.013657 | 0.135169 | 0.144129 |
| **pdays** | -0.034381 | -0.047556 | 0.052606 | 1.000000 | -0.587508 | 0.271063 | 0.078920 | -0.091374 | 0.296946 | 0.372659 |
| **previous** | 0.024379 | 0.020600 | -0.079182 | -0.587508 | 1.000000 | -0.420587 | -0.203197 | -0.050929 | -0.454571 | -0.501411 |
| **emp_var_rate** | -0.000242 | -0.027941 | 0.150786 | 0.271063 | -0.420587 | 1.000000 | 0.775293 | 0.196257 | 0.972244 | 0.906949 |
| **price_index** | 0.001009 | 0.005303 | 0.127826 | 0.078920 | -0.203197 | 0.775293 | 1.000000 | 0.059170 | 0.688180 | 0.521945 |
| **confidence_index** | 0.129075 | -0.008126 | -0.013657 | -0.091374 | -0.050929 | 0.196257 | 0.059170 | 1.000000 | 0.277864 | 0.100679 |
| **eur_3month** | 0.010852 | -0.032861 | 0.135169 | 0.296946 | -0.454571 | 0.972244 | 0.688180 | 0.277864 | 1.000000 | 0.945146 |
| **no_of_employees** | -0.017607 | -0.044672 | 0.144129 | 0.372659 | -0.501411 | 0.906949 | 0.521945 | 0.100679 | 0.945146 | 1.000000 |