# Phase 1

- Business requirements detailed in phase 0

# Exploratory Data Analysis (EDA) - customer support

## Main objective with this notebook -> starting point to identify what drives customer churn

- Load the dataset and and analyse some features
- Evaluate some historical information and characteristics of the data
- Run one machine learning algorithm (Ensemble: Random Forest in this example) and evaluate the current data
- Plot the most important features / characteristics that drives the decision to churn or not

#### Dataset

- the dataset used in this process can be accessed trought IBM website below

  https://www.ibm.com/communities/analytics/watson-analytics-blog/guide-to-sample-datasets/ (https://www.ibm.com/communities/analytics/watson-analytics-blog/guide-to-sample-datasets/)

In [1]:
```
## Import libraries used in this notebook
import pandas as pd
import pandas_profiling as pp
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

In [2]:
```
df = pd.read_csv('../data/WA_Fn-UseC_-Telco-Customer-Churn.csv')
df.head(5).T
```

Out[2]:

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **customerID** | 7590-VHVEG | 5575-GNVDE | 3668-QPYBK | 7795-CFOCW | 9237-HQITU |
| **gender** | Female | Male | Male | Male | Female |
| **SeniorCitizen** | 0 | 0 | 0 | 0 | 0 |
| **Partner** | Yes | No | No | No | No |
| **Dependents** | No | No | No | No | No |
| **tenure** | 1 | 34 | 2 | 45 | 2 |
| **PhoneService** | No | Yes | Yes | No | Yes |
| **MultipleLines** | No phone service | No | No | No phone service | No |
| **InternetService** | DSL | DSL | DSL | DSL | Fiber optic |
| **OnlineSecurity** | No | Yes | Yes | Yes | No |
| **OnlineBackup** | Yes | No | Yes | No | No |
| **DeviceProtection** | No | Yes | No | Yes | No |
| **TechSupport** | No | No | No | Yes | No |
| **StreamingTV** | No | No | No | No | No |
| **StreamingMovies** | No | No | No | No | No |
| **Contract** | Month-to-month | One year | Month-to-month | One year | Month-to-month |
| **PaperlessBilling** | Yes | No | Yes | No | Yes |
| **PaymentMethod** | Electronic check | Mailed check | Mailed check | Bank transfer (automatic) | Electronic check |
| **MonthlyCharges** | 29.85 | 56.95 | 53.85 | 42.3 | 70.7 |
| **TotalCharges** | 29.85 | 1889.5 | 108.15 | 1840.75 | 151.65 |
| **Churn** | No | No | Yes | No | Yes |

| | customerID | gender | SeniorCitizen | Partner | Dependents | tenure | PhoneService | MultipleLir |
|---|---|---|---|---|---|---|---|---|
| **0** | 7590-VHVEG | Female | 0 | Yes | No | 1 | No | No phone serv |
| **1** | 5575-GNVDE | Male | 0 | No | No | 34 | Yes | No |
| **2** | 3668-QPYBK | Male | 0 | No | No | 2 | Yes | No |
| **3** | 7795-CFOCW | Male | 0 | No | No | 45 | No | No phone serv |
| **4** | 9237-HQITU | Female | 0 | No | No | 2 | Yes | No |

## Prepare the data to run Random Forest Classifier

- and also drop the information customer_id  (will not be used)

In [4]:
```
drop_items = ['customerID']

df.drop(drop_items, axis = 1, inplace = True)
```

In [5]:
```
# LabelEncoder
le = LabelEncoder()

# apply "le.fit_transform"
df = df.apply(le.fit_transform)
```

In [6]:
```
target = 'Churn'
features = df.columns.tolist()
features.remove(target)

X = df[features]
y = df[target]
```

## Run Random Forest and print the best score

In [7]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=12345, stratify=y)


## Import the random forest model.
from sklearn.ensemble import RandomForestClassifier
## This line instantiates the model.
fit_rf = RandomForestClassifier()
## Fit the model on your training data.
fit_rf.fit(X_train, y_train)
## And score it on your testing data.
fit_rf.score(X_test, y_test)


print('----------------------- The best score/accuracy is: ' + str(fit_rf.score(X_test, y_test)))
fit_rf.score(X_test, y_test)
```

```
----------------------- The best score/accuracy is: 0.7771469127040455
```

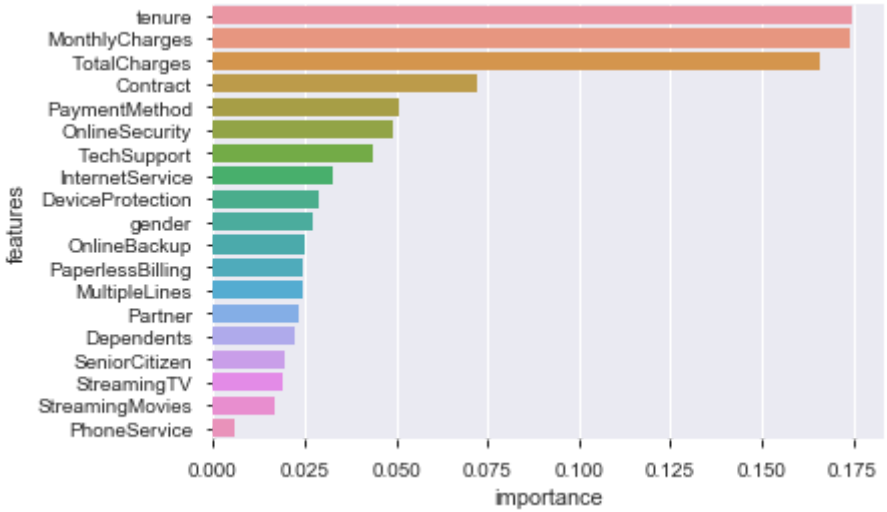
Out[7]:
```
0.7771469127040455
```

## Plot Feature Importance

In [8]:
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# Plot feature importance
feature_importances = pd.DataFrame({'features': X_train.columns,
                                    'importance': fit_rf.feature_importances_}).sort_values('importance', ascending=False)

ax = sns.barplot(x="importance", y="features", data=feature_importances)
```



# Summary

## The 5 most important features initially identified and related to Churn (Yes - No) are:

- Monthly Charges
- Tenure
- Total Charges (and also correlated to Monthly Charges and tenure)
- Contract and
- Payment Method

  all of these information can be seen in the graphic above

## 26.5% of these customer_base are churners (current dataset) and this churn rate is to high

## An strategic plan could be designed to decrease this churn rate