Phase 2 - build machine learning models with Python

- The model with best results (higher RECALL metric) was the GBM using sklearn
- Detailed information could be seen in previous notebooks - build machine learning models with Python - 1 to 5
- This notebook will provide the final python script for deployment along with rules to run the GBM machine learning model

Sample execution of new prediction - execution of 1 python script with new_data as a .csv file

Detailed data pipeline process

- Load the GBM model export in the last notebook
- Load the statistics information applied during the GBM model build
- Apply the data pipeline rules to new data (standard scale and one hot encode for categorical features)
- Run predictions and check the results

The cell below just show all code in the python script - python_prediction_customer_churn.py that executes the prediction

- Load the script with command %load python_prediction_customer_churn.py
- All lines were commented

In [1]:

```
# %load python_prediction_customer_churn.py


## python script
## python_prediction_customer_churn.py <dir/file_name>

# ## Load ML models
# import pickle
# import pandas as pd
# import os
# import sys

# # a = os.getcwd()
# def f_validation_process():
#     print()
#     print('---- Error..')
#     print('-----------  Run the program with valid file name for new predictions')
#     print()
#     print('python python_prediction_customer_churn.py <dir/filename>')
#     print()

#     raise ValueError('Re-start the process with correct parameter:filename.')

# if (len(sys.argv) != 2):
#     f_validation_process()

# ## Load GBM Model - prod model
# file_export_model = './ML_models/model_GBM_prod_v1.sav'
# gbm_prod_model = pickle.load(open(file_export_model, 'rb'))

# # ## New Data for prediction
# # df_new_prediction = pd.read_csv('./ML_models/pred_baseline_example.csv')

# # df_new_prediction.head(3)

# ## check if file exist and load the data
# # path_pred_file = './ML_models/new_data_customer_churn.csv'
# path_pred_file = sys.argv[1]

# if not(os.path.isfile(path_pred_file)):
#     print('Inform a correct file path. Sample: ./data/file_name.csv')
#     raise('Invalid file name')
# else:
#     print('Processing file...')
#     df_new_prediction = pd.read_csv(path_pred_file)
#     df_export_prediction = df_new_prediction.copy()
#     # df_new_prediction.head(2)

# ## Load statistics information from training data during the ML build and cols orders
used for prediction
# stats = pd.read_csv('./ML_models/stats_df_train.csv', index_col=0)
# cols_prediction = stats.columns.to_list()
# cols_std = ['tenure', 'MonthlyCharges' , 'TotalCharges'] ## specific cols to apply sc
ale
# stats = stats.T
# stats = stats[stats.index.isin(cols_std) ]

# ## StandardScale_x aux function
# def std_scale_x(x):
```

```
#     """ This function will standard scale based on training data
#         Only specific cols ['tenure', 'MonthlyCharges' , 'TotalCharges']
#     """
#     return (x - stats['mean']) / stats['std']

# ## Apply all rules in the data and run the prediction
# std_cols = ['tenure', 'MonthlyCharges' , 'TotalCharges']
# cat_cols = [col for col in df_new_prediction.columns.to_list() if col not in std_col
s]

# ## Apply scale
# df_new_std_cols = std_scale_x(df_new_prediction[std_cols])

# ## Apply OHE and merge the data
# df_categorical = df_new_prediction[cat_cols]
# df_new_pred2 = pd.concat([df_new_std_cols, df_categorical], ignore_index=False, axis=
1)
# df_new_pred2 = pd.get_dummies(df_new_pred2, columns=cat_cols)

# ## Include all columns missing to run the prediction
# cols_OHE = [x for x in cols_prediction if x not in df_new_pred2.columns.to_list()]
# for col in cols_OHE:
#     df_new_pred2[col] = 0

# df_new_pred2 = df_new_pred2[cols_prediction]

# ## Sample data for prediction
# # df_new_pred2

# # Run prediction and check the result
# X_prediction = df_new_pred2.values
# y_pred_gbm_prod = gbm_prod_model.predict(X_prediction)
# prediction_result = lambda x: 'Churn' if x == 1 else 'No Churn'
# # print('Sample of 1 Customer prediction: ', prediction_result(y_pred_gbm_prod))
# # y_pred_results = prediction_result(y_pred_gbm_prod)
# df_new_pred2['Churn_prediction'] = y_pred_gbm_prod
# df_export_prediction['Churn_prediction'] = df_new_pred2['Churn_prediction'].apply(pre
diction_result)

# ## export results
# df_export_prediction.to_csv('./ML_models/Churn_prediction_results.csv', sep=',', inde
x_label=False)
# print('Prediction done!')
# print('  - Check results in the file: Churn_prediction_results.csv')
```

Running new predictions - customer churn

- Deployment done with 1 single script : python_prediction_customer_churn.py
- Inform the file name as a parameter for the script

The example below show the execution and the results provided

In [2]:

```
%%time
!python python_prediction_customer_churn.py ./ML_models/new_data_customer_churn.csv
```

```
Processing file...
Prediction done!
  - Check results in the file: Churn_prediction_results.csv
Wall time: 2.51 s
```

Churn predicitons generated in the .csv file

In [3]:

```
!head -n 3 ./ML_models/Churn_prediction_results.csv
```

```
tenure,MonthlyCharges,TotalCharges,gender,PaymentMethod,Contract,Churn_pre
diction
185,1,24.8,24.8,Female,Electronic check,Month-to-month,Churn
2715,41,25.25,996.45,Male,Bank transfer (automatic),Month-to-month,No Chur
n
```

Summary

- With the information provided in this Advanced Analytics process many actions can be implemented to retain these customers that will probably churn
- The next and final notebook will show some additional considerations

In [ ]:

```

```