# AXA-Challenge

machine learning project

## Feature Engineering

### feature_engineering1.r

Input: databrute.csv (select DATE, DAY_WE_DS, ASS_ASSIGNMENT, CSPL_RECEIVED_CALLS from train_2011_2012_2013.csv)

Output: **rawdata** folder, containing seperate data for every ASS_ASSIGNMENT.
For every file, we have format:
mydate(Y-M-D), mymonth(1-12), myday(1-31), DAY_WE_DS(1-7), hour_index(1-48), CSPL_RECEIVED_CALLS

### feature_engineering2.r

Input: **rawdata** folder

Output: **data_v1**, **data_v2**, **data_v3** folder with training data.
**pred_data_v1**, **pred_data_v2**, **pred_data_v3** folder with data that needs to be predicted.

v1: normalized features, with new features: **same_hour_seven_days_ago**, **same_day_seven_days_ago**, **mean_value_last_week**.
v2: based on v1, change all periodic features to its **sin** and **cos**.
v3: based on v1, change all periodic features to its $\mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4$

## SVR

### svr.py + linearKernel.py

Input: **data_v1/v2/v3**

Output: too long to compute

# Linear regresstion

## training.py or training_parameter.py

Input: **data_v1/v2/v3**

Output: **training_parameters_v1/v2/v3**

## predicting.py

Input: **training_parameters_v1/v2/v3**

Output: **pred_result_v1/v2/v3**

## submission.r

Input: **pred_result_v1/v2/v3**

Output: submission.txt

# Gradient boost tree

## format_treeinput.r

transform data format of **data_v1/v2/v3** to libSVM format

## gradient_boost.py

Input: transformed **data_v1/v2/v3**

Output: **pred_result_v1/v2/v3**

## submission.r

Input: **pred_result_v1/v2/v3**

Output: submission.txt