# HW3 (due date: 2021 01 22)

IMPORTANT: For grading to be done correctly make sure you follow the instructions precisely.

Deliverables: You are required to submit the following two files:

1. A Python file containing all the commands and scripts.
2. A pdf file summarizing what you have done and the results for each exercise; basically, your report. The pdf file should contain only the essentials. Up to one-third of the assignment grade can be deducted for poorly organized and formatted report.

**Q1)**

For this exercise we will use the data stored in **dats501\_hw3\_dataForLag**.**xlsx**.

We have an A variable that uses its older values for prediction (auto regressive). We name the current hour as T. While trying to predict T+1 which we will say A\_future1, we have closest T-1 value, A\_lag1. The values of B and C are also available for T-1 and you are also expected to create their lagged values since we think their values may also be effective to predict A

1. We want to create suitable data to predict T+1, T+2, T+3 moments. Create loops to create below:
   1. 2 new lagged columns for column A (from lag1 to lag3)
   2. 4 new lagged columns for column B (from lag1 to lag5)
   3. 6 new lagged columns for column C (from lag1 to lag7)
   4. 3 new lead columns for column A (from future1 to future3)

The naming will be like “'A\_lag' + str(j)” for the new lagged columns.

Note that: Some initial ours cannot be used since leading and lagging operations create some NULL values since the values are not available

1. Append those data frames into a list in the loop. Do not make appending manually outside the outer loop.

**Q2)**

Create data with below code

“import pandas as pd

import numpy as np

from sklearn.datasets import load\_breast\_cancer

bc = load\_breast\_cancer()

X,y = pd.DataFrame(bc.data), bc.target

X[ 'cat1' ] = np.random.choice( a=['cx', 'cy', 'cz'], size=569, p=[0.5, 0.3, 0.2] )

X[ 'cat1' ][0:100] = y[0:100]

X[ 'cat1' ][300:500] = y[300:500]

X[ 'cat2' ] = np.random.choice( a=['x', 'y', 'z'], size=569, p=[0.2, 0.2, 0.6] )

X[ 'cat2' ][100:200] = y[100:200]

X[ 'cat2' ][250:400] = y[250:400]

X[ 1 ][1:10] = np.nan

del bc“

1. Use scikit-learn versions of below ml methods. You do not need to optimize parameters:
   1. Linear regression
   2. Lasso
   3. Ridge
   4. Elastic net
   5. Logistic regression
   6. Nearest neighbours
   7. Naïve Bayes
   8. Classification decision tree
   9. Bagging
   10. Random forest
   11. Extra trees
   12. Adaptive boosting
   13. Gradient boosting
   14. Catboost
   15. Lightboost
   16. XGboost
2. Make variable selection with
   1. SelectKBest
   2. RFE
   3. SelectFromModel
3. Make predictions with the selected variables. For the predictions use best 4 algorithm (it should contain at least a linear model and at most 2 boosting) found in the first step
4. Decide which variables are best and use 3 best models to use in
   1. sklearn.ensemble.VotingClassifier
   2. sklearn.ensemble.StackingClassifier