# Graded Assignment 2 | DTE-2501 | H22

Good news! The engine you designed for AFC using genetic algorithms is a roaring success! As a result, rumors of the power of AI have reached the big wigs of AFC, and they see a new potential use case. In addition to aircraft engines, AFC also happens to be one of the top suppliers of shower curtain rings and shower curtain ring accessories. However, AFC has never been able to meet demand, and always end up with either too much or too little inventory for any given day. Being AFC's official expert of applied AI, you know that *regressions* can be used to create a forecasting model. You also know that you can combine simpler models into an *ensemble* in order to improve performance. You are given two sets of data; one set contains the demand records from last year, and the other contains the demand so far this year. Using this data, founder and CEO of AFC, Elgen Tusk, has asked you to make a system to forecast and visualize demand for the remaining year.

## Task 1 (max 5 points)

Identify and break down the problem in the context of regression and ensembles. Describe briefly how you are going to attack it (2-3 sentences in the report).

## Task 2 (max 20 points)

Create a decision tree regressor by following each step a-c, and describe briefly in the report how you solved each step:

a) Implement a function to read the file "data_2021.csv" and create a data set for a regression tree. Use the sliding window approach (6 pts)
b) Create a function to train a single regression tree on the dataset you just created (5 pts)
c) Using the same sliding window approach as in (2.a), start from 01.01.2021 and use the regression tree to predict the next value. Save each prediction in a list. Use *matplotlib* to plot the results alongside the original data. Comment on your findings in your report (9 pts)

## Task 3 (max 40 points)

Create your own implementation of a random forest by following each step a-d. Describe briefly in the report how you solved each step:

a) Based on the tree you implemented in (2), create an ensemble of regression trees. Decide on the number of trees, you can for example start with 50-100 trees. Each tree should be initialized with random parameters. (10 pts)
b) Implement *bootstrapping*, and train each of your trees on a random part of the dataset (15 pts)
c) Implement *aggregation* to combine the output of all your trees to a single value (5 pts)
d) Using the same sliding window approach as in (2.a), start from 01.01.2021 and use the ensemble to predict the next value. Save each prediction in a list. Use *matplotlib* to plot the results alongside the original data and your predictions from (2.c) (10 pts)

## Task 4 (max 20 points)

A multi-layer perception is another simple learner that may be used in an ensemble. Follow the steps a-c. and describe briefly in the report how you implemented each step:

a) Create an ensemble of *MLPClassifier*s from *sklearn.neural_network* (5 pts)
b) Train the new ensemble you just created using bagging (5 pts)
c) Using the same sliding window approach as in (2.a), start from 01.01.2021 and use the ensemble to predict the next value. Save each prediction in a list. Use *matplotlib* to plot the results alongside the original data and your predictions from (2.c) and (3.d) (10 pts)

## Task 5 (max 15 points)

Now that you have created and trained a selection of models, it is time to put them to the test and predict this year's demand. Follow the steps a-c, and describe briefly in the report how you implemented each step:

a) Load the file "data_2022.csv" and create a test set in the same format as the training set you created in (2.a) (5 pts)
b) Pick a model from (2, 3, or 4), and use this year's demand so far to predict the missing values from 26.08.2022 to 31.12.2022. Save the predictions in a list (5 pts)
c) Use *matplotlib* to plot the predictions for the remainder of 2022 (5 pts)