# Ensemble

Graded assignment 02 - report by Frédéric Charon

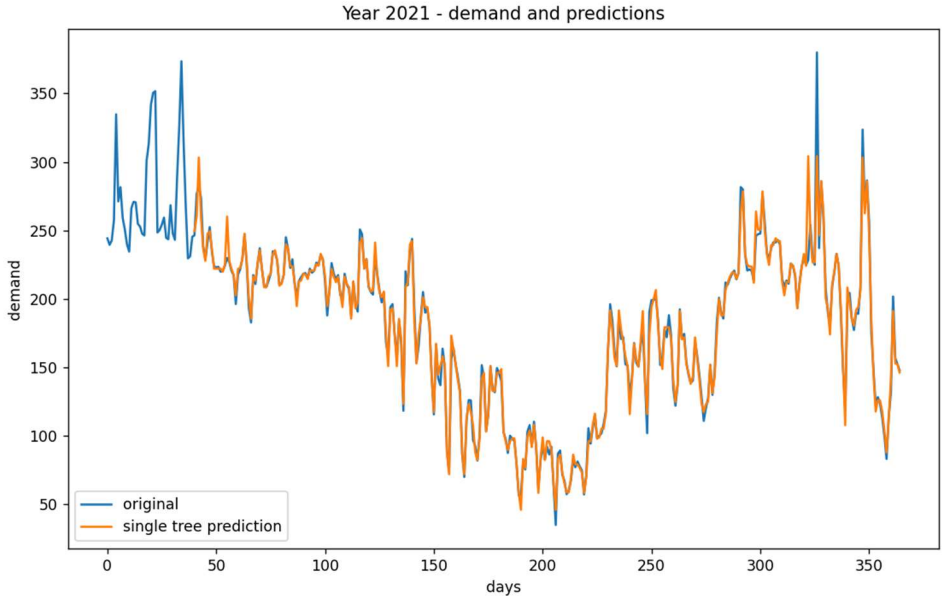
Task 1)

For this assignment we are given two data files, one containing the demand of a product in 2021 for the whole year and one for 2022 from January to August. Our task is, to use that data to predict and forecast the demand for the remaining months of the year 2022 by using regression and ensembles. To solve this task, I am going to create an ensemble of regression trees using bagging and the sliding window approach, training it on the data from 2021 and finding the best performance to use the year 2022 as test data and forecasting the unknown demands.

Task 2)

First, we're going to create the dataset using the sliding window approach. We set a window size w and iterate through the data by assigning to each datapoint starting from index w an input containing the w previous data points.

After our dataset is finished, we can create our ensemble. For now, we start by creating a single tree and predicting all the outputs, then we plot them alongside the original demands:



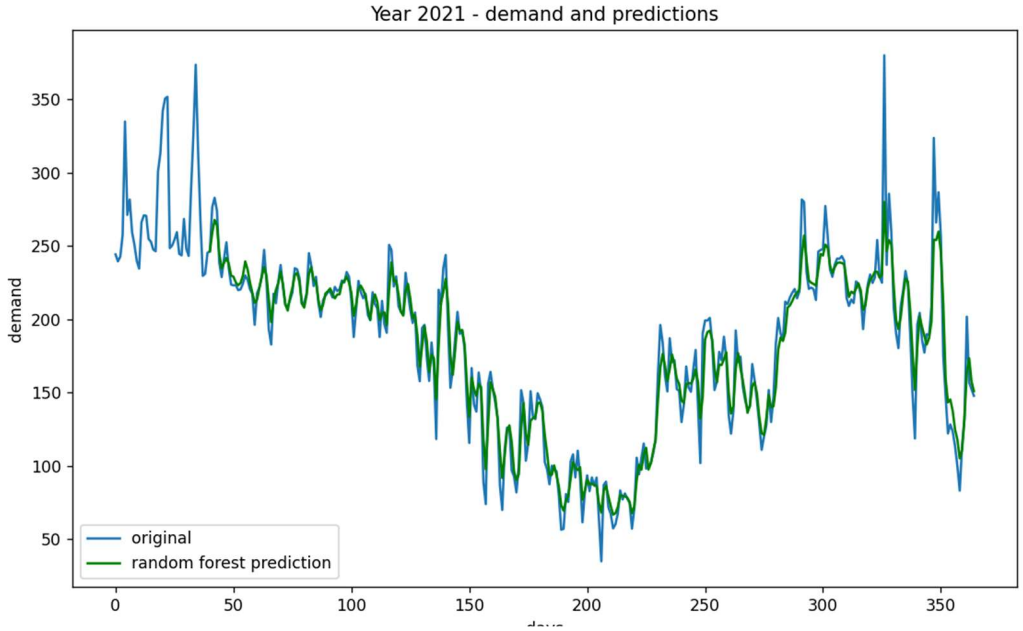

We can see that the tree is performing pretty good on the data, it doesn't fit perfectly but it follows the same trends as the original – however this good performance is not very surprising since we used the same data for training and testing. Our goal is to have a good fitting prediction in the end (not generalizing too much), but if it fits too good, we might run into overfitting. The predictions start at day 40, this is due to the window size of 40.

Task 3)

Now, that we implemented the function to create a tree, we can create any number of trees and combine them into an ensemble. We randomize the parameters of each tree to create a random forest, then we become able to get many predictions and calculate the mean value – the mean value of an ensemble is in general more reliable than the prediction of a single tree.

Instead of training each tree on the whole dataset we use bootstrapping, this means that each tree is only trained on a random subset of the original data which makes each tree even more unique.

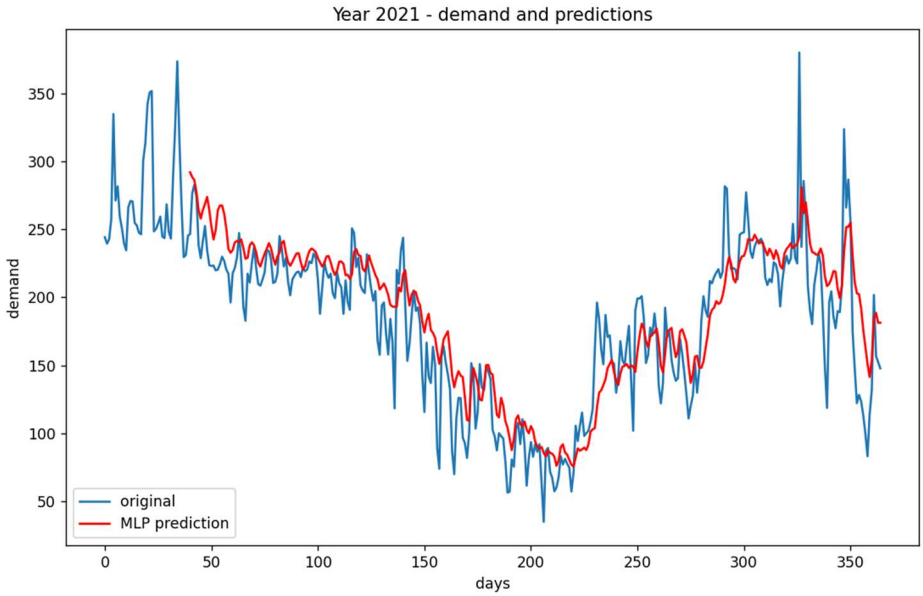
We combine bootstrapping with aggregation (-> bagging), which means that we combine the predictions of the ensemble to get our final value. In the code this is done in the make_prediction() function, we combine the predictions by taking the mean value.




After plotting the random forest, we can see that it almost behaves the same way as the actual demand, the general shape looks the same, but it is smoothing out the peaks a bit. Compared to the single tree in 2) this is more remarkable since no tree was trained on the whole dataset, they all used only a random subset instead.

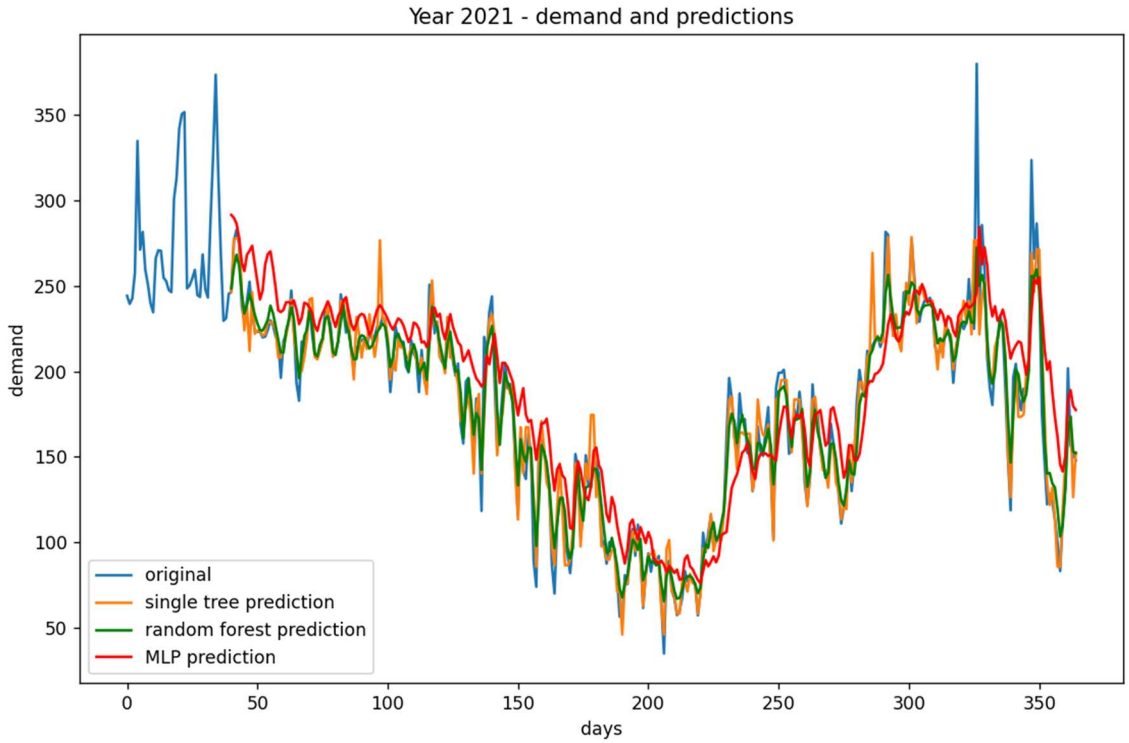
Task 4)

This task is similar to the previous one but instead of a random forest we create an ensemble of multi-layer perceptrons (MLPs). We use the same number of classifiers and the same bagging and sliding window functions.




The result here is less accurate than the one of the random forest, it is smoothing out the peaks even more and the shape looks less similar, it looks more like an approach to show the general trend than the explicit values.

Here's the plot of all the predictions at once:



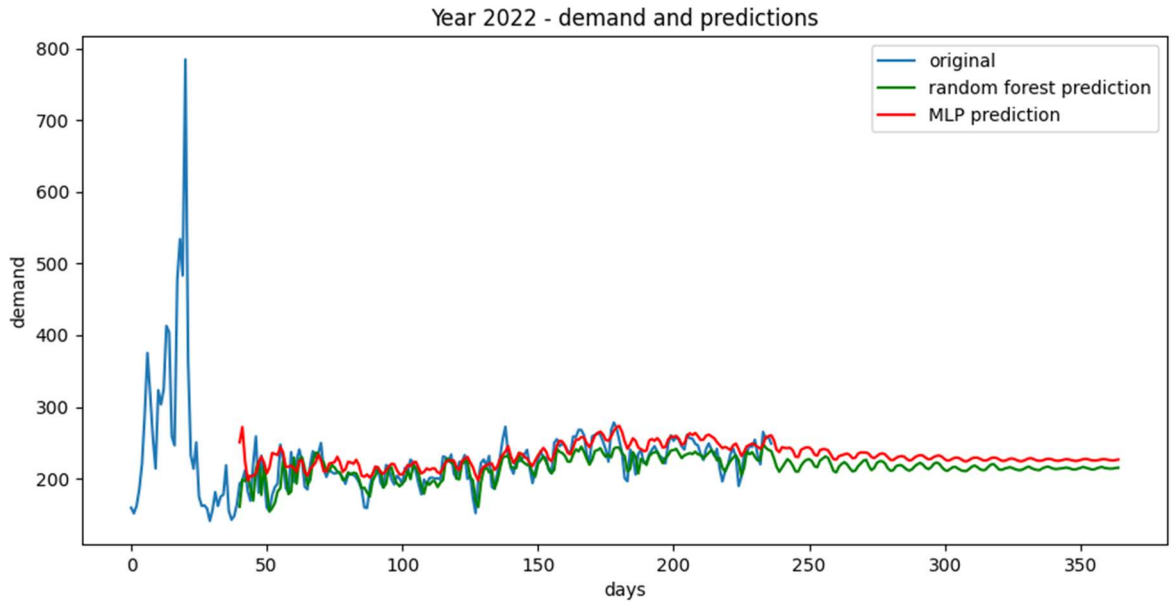

Comparing the predictions, we can see that the single tree is not always having the same shape as the original and a bit off at some points, the predictions seem to randomly under- or overfit the actual values sometimes.

The random forest has almost the same shape as the original, but it doesn't really match the optimums, it smooths out all the peaks.

The last one, the MLP prediction, seems to generalize the behaviour of the predictions more than the others, the shape is rather showing the long-term trend by smoothing out the peaks even more and the shape is not as accurate as for the random forest. So for this data the random forest looks the most promising, let's see how they perform on the test data.

Task 5)

Finally, we predict the data for the year 2022, we have data points until August and we want to forecast the demands for the rest of the year. As long as there are datapoints in our dataset I used them as input value for the predictions, for the data points with not enough input data we are using the predicted values from the previous dates. We do this until we get the data for the whole year 2022.




In this case I decided to plot both, the random forest and the MLP regressor, it is important to use the exact ensembles from the training process. Both predictions result in a gradient fade since both ensembles are continuously smoothing out the peaks in each iteration after there is no input data anymore and they must use the last predicted values as new input data.

Conclusion:

Ensembles are used to predict and forecast future values based on a given dataset. We compared two different ensembles, for this kind of data the random forest seems to perform better than the MLP since the predictions are closer to the actual values.

We also see some challenges for this approach, even the random forest doesn't predict very accurate values because of the smoothing. Especially when predicting a lot of data without previous inputs, then the data just flattens in form of a gradient fade or might get suck in a loop where it predicts the same behaviour again and again.