|  |  |  |  |
| --- | --- | --- | --- |
|  | | **PTS 2022** | |
| Previsão de Séries Temporais  Project | | | |
| **Team nr:** 4 | **Student 1 :** Miguel Figueiredo | | **IST nr:** \_\_\_\_\_\_\_\_\_ |
| **Student 2 :** João Costa | | **IST nr:** \_\_\_\_\_\_\_\_\_ |

# Data Profiling

## Data Dimensionality

* **SP500:** The dataset has 4 variables for 11.193 time-points (weekly data from 03/01/1978 – 20/05/2022), so it does not seem to present a data dimensionality problem, so no balancing seems to be needed.
* **Pollution:** The dataset is comprised of 17 variables and 35.064 time-points (hourly data from 2013-03-01 00:00:00 to 2017-02-28 23:00:00). Once again, there is no data dimensionality issues.

## Data Granularity

* **SP500:** All the different granularities from weekly to yearly timespans clearly show an ascending trend for SP500 close price (target variable), not differing from the granularity chosen (as we can see from the Appendix images).
* **Pollution:** Looking at the three granularity graphs (in the Appendix) it is not possible to find a clear trend over the timespan of the dataset. Nevertheless, there seems to be a cyclical pattern that repeats itself every year (e.g., high PM2.5 concentration (our proxy for air pollution) around the beginning of each year).

## Data Distribution

* **SP500:** As we increase the granularity from weekly to yearly, we can observe from the boxplots a diminishing number of outliers, which is expected as there are fewer data points as we increase the granularity (see Appendix). Moreover, the outliers observed in the given granularity (weekly) are explained by the fact that the series continues to increase along the timeseries and exponentially grows in the final years, so they are seen as outliers in the series. Looking at the histograms over differing bins (10, 25 and 100), we can see that the distribution observed seems to be coherent for the differing bins and time aggregations considered, with more data points closer to the lower values observed.
* **Pollution:** The boxplots of PM2.5 concentration with different granularities (from hourly to quarterly),(see Appendix), exhibit the expected trend: more outliers the more granular is the distribution and no outliers with yearly granularity. Additionally, once there is a natural lower bound for this variable, zero, the outliers tend to be greater values and not so much small ones.

## Data Stationarity

* **Graphical user interface, chart

  Description automatically generatedSP500:** Following the past observations over the data distribution for the different granularities, we can further confirm that the series in non-stationary as we observe an increase both in its average and variance over the last half of the timeseries, so that some corrections (differentiation for example) might improve the results of the models applied.
* **Pollution:** The analysis of the data stationarity is not straightforward in this case. Looking at the daily series, we notice that the spikes seem to be a little bit higher each year, however, taking the mean by year we notice a somewhat stationary behavior along the time. Below are two graphs that take the mean for windows of different sizes: 10 bins in the first and 5 bins (more or less an yearly mean).

**Chart, line chart

Description automatically generated**

**Chart, line chart

Description automatically generated**

# Chart, bar chart Description automatically generatedData Preparation

* **SP500:** After applying the necessary treatments for missing values and scaling differences (standard scaler utilized), we observe that the best preparation using the MLP model (Multi Layer Perceptron) has a performance differentiator for comparing the different preparation techniques used. As a base model for performance comparison, we ran the MLP without any of the preparations mentioned, yielding a R2 = 0.67.

Chart, bar chart

Description automatically generated

* **Pollution:** After missing values imputation and standard scaling the MLP model was used to evaluate and serve as benchmark to evaluate the benefit of the transformations. The R2 obtained was 0.88.

## Aggregation

* **SP500:** As we can observe (see Annex), there is a decrease from R2 = 0.67 from the base model to R2=0.54, 0.63 and 0.25 to monthly, quarterly and yearly aggregations accordingly. Thus, we decided not to proceed with any aggregation of the data, as it significantly decreases the performance of the model.
* **Pollution:** As it becomes apparent by looking at the graphs in the Annex, the alternative that yields the best results with aggregation is the Daily aggregation (R2=0.93), followed by the model with weekly aggregation (R2=0.90) and the model with no aggregation applied (R2=0.89). Default is hourly in the dataset. Other aggregations were also tested but the results were poorer. Given this, we decided to proceed with the daily aggregation.

## Differentiation

* **SP500:** Applying differentiation over the aggregated dataset increases the performance of the model to an R2 = 0.73, so we decided to proceed with this transformation, conforming to initial analysis that it seemed to be beneficial to the dataset due to its non-stationarity (as we can observe, the stationarity of the timeseries seems to be increased, although variance still seems to vary over the timeseries, with more variation towards the most recent dates).

Chart

Description automatically generatedChart, bar chart

Description automatically generated

* **Pollution:** The application of differentiation decreases the performance of the model R2=0.86 < R2=0.93). Thus, we do not apply this transformation moving forward.

## Smoothing

* **SP500:** For 10 bins the performance decreases to R2=0.21 and with 100 bins the R2 turns negative to -7.65, so we made decision of not applying the smoothing technique to the timeseries (see Annex).
* **Pollution:** The performance for 10 bins is R2=0.89, whereas for 100 bins is R2=0.92 (see Annex). Even though the results are encouraging, they represent a decrease in performance taking in consideration the R2=0.93 attained without smoothing. For that reason, we decided not to apply smoothing.

# Prediction

## KNN

* **SP500:** As we can observe, KNN distances and weight methods performed similarly over the several K number of neighbours tested. Nonetheless, the best results were achieved with a uniform weight and using Manhattan distance for 19 K neighbours, which yielded a R2 of 0.92, therefore increasing the performance of our model considerably. We also observe a higher R2 for the testing dataset than the training one (R2=0.67), which is surprising. This could be due to the data not being independent as we supposed, so additional searching and suited treatments could have been made to better understand this result.

Graphical user interface, application

Description automatically generated

We also see an overfitting phenomena from around K=13, since for lower K’s the training dataset exponentially grows to a R2 of 1, while the test dataset decreases progressively to around a R2 of 0.65. This is normal since a K=1 means that the predicted values rely simply on its nearest neighbour, which will be different in the training and testing dataset, resulting in the overfitting phenomena we observe. Since we chose a K=19, we are out of the overfitting area, and thus we yield better and more solid results.

Chart, line chart

Description automatically generated

* **Pollution:** Both KNN distances and weight methods had similar performances, as shown by the graphs bellow, for the various k-neighbors tested. Nonetheless, the best results were achieved with a distance weight and using Euclidean distance for 5 neighbors, which yielded a R2 of 0.92

Graphical user interface, application

Description automatically generated

Furthermore, the training achieves maximum performance and the test results (the relevant ones) suggest the model enters in overfitting for k<4, maybe even with a greater k, k<5.

Chart, bar chart

Description automatically generatedGraphical user interface

Description automatically generated

## Regression Trees

* **SP500:** Regression trees seem to have a lower result than KNN in general, although the best regression tree model yielded a R2=0.88, which is still high. This model was achieved with a Friedman MSE criteria, a minimum impurity decrease of 0.0001 and a tree depth of 15 levels. We observe that MSE criteria remains flat with an R2 of approximately 0 for all impurity decrease levels and depth levels. This difference might ensue since Friedman’s MSE splitting criterion chooses the split not only based on how close we are to the target (which is MSE approach) but also based on the probabilities of the desired region split that we are trying to find. The MAE criteria only presents good results for a minimum impurity decrease of 0.0001 and only for 7 or more levels of depth.

A picture containing text, screenshot, computer

Description automatically generated

We are also able to conclude that there are indeed overfitting phenomenon starting at a maximum depth of around 15 / 17 levels, since the train R2 increases almost to 1 but the test R2 starts decreasing until below 0.7.

Chart, line chart

Description automatically generated

* **Pollution:** The best regression tree model was achieved with absolute error criteria, a minimum impurity increase of 0.001, depth of 5. The R2 obtained was 0.92. Additionally, we observe that for depths greater than 5 the best model is overfitted as shown in the graphs.

Chart, line chart

Description automatically generatedChart, bar chart

Description automatically generatedGraphical user interface, application, line chart

Description automatically generated

## Random Forests

* **SP500:** The best Random Forests model yielded a R2=0.93, slightly increasing the performance compared to the KNN model. This model was achieved using a maximum of 75% of the total features available, a maximum depth of 15 (in line with the maximum depth observed for our best single regression tree model) and 200 estimators. As we can see below the best models tend to be the ones with higher depths, and as the number of estimators increase also does the performance. Moreover, there’s a huge discrepancy in the performance of the models using only 5 and 10 depth levels, but not so much from 10 to the max depth tested (25), suggesting that the model could be more relaxed in this feature trading complexity by a faster and simpler model only decreasing the performance marginally. Comparing the usage of 25% and 75% of the total features available, we see a bigger difference when using less estimators, but the difference tends to decrease when using more estimators for the training, bridging the gap between the two methods.

A screenshot of a computer

Description automatically generated with medium confidence

When using Random Forests the overfitting phenomena appears to have disappeared when comparing with single Regression Trees since the performance seems to follow a straight line (although there’s some fluctuation according to the number of estimators used).

Chart, line chart

Description automatically generated

* **Pollution:** The best Random Forests model yielded a R2=0.94, which is a better performance compared to the models seen so far. This model was achieved using a maximum of 75% of the total features available, a maximum depth of 20 and 75 estimators.

Graphical user interface, line chart

Description automatically generated

Chart, bar chart

Description automatically generatedLine chart

Description automatically generated

## Gradient Boosting

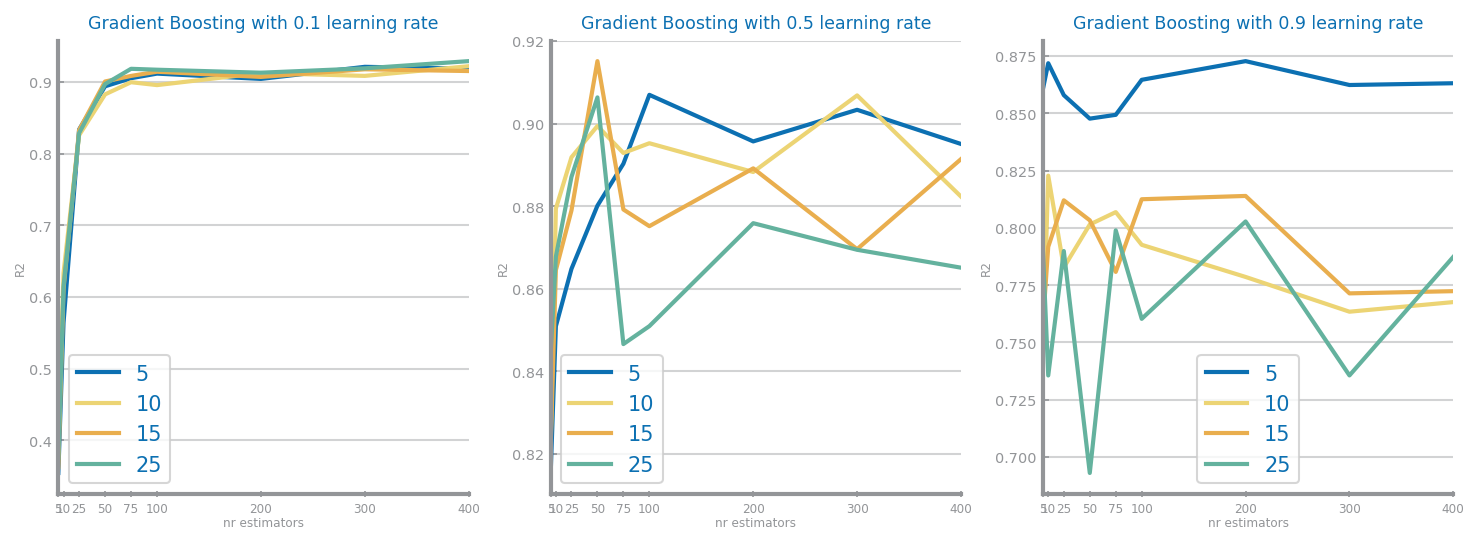
* **SP500:** The best Gradient Boosting model achieved a R2=0.97, substantially increasing our performance. This was achieved with a learning rate of 0.1, a depth of 5 and with 300 estimators. We consistently observe better results for a lower learning rate of 0.1, so we can state that this is one of the most important parameters chosen. Moreover, the depth level seems also to be relevant across all learning rate levels and number of estimators, with lower depths having better performance results, indicating that lower depth levels are relevant for the task at hand. Regarding the number of estimators we only observe higher performance according to more estimators for a learning level of 0.1, but not for higher learning rates, so we believe that the number of estimators is not as relevant as the other parameters identified.

A screenshot of a computer

Description automatically generated with low confidence

Overfitting also does not seem to be a problem, since we observe a flat straight line for both test and train datasets as we increase the number of estimators. (Same for the Pollution dataset – see Annex)

* **Pollution:** The best Gradient Boosting model achieved a R2=0.93. This was achieved with a learning rate of 0.1, a depth of 25 and with 400 estimators. The learning rate of 0.1 seems to be the most relevant factor determining performance as commented above. The fact that the greater depth tested for *lr* =0.1 (i.e. depth=25) yields the best results goes is not shared by any of the other models.



## Multi-Layer Perceptrons

* **SP500:** As observed in the differentiation step (since we used MLP as a basis for performance comparison of the different timeseries transformation methods employed), we achieved a R2=0.73 for a constant learning rate of 0.1 and 150 maximum iterations. We once again observe that lower learning rates provide better and more sustained results along the number of iterations performed. Moreover, a learning rate of 0.9 completely decreases the performance of the model to very negative R2 levels.

We also do not seem to observe overfitting phenomenon as the number of episodes to train the model increase, although we do observe a valley shaped performance around 500 episodes that decreases the R2 from its highest to -1. Most importantly, the lower the number of episodes the more unstable the model, since the first episodes R2 range from -1 to -4.5, far from the desired performance.

* **Pollution:** The best model (R2=0.94) was achieved with *invscaling*, learning rate of 0.1 and maximum iterations of 150. There is no evidence of overfitting phenomenon since the test R2 monotonically increases with the number of episodes.

**A screenshot of a computer

Description automatically generated with low confidenceChart, bar chart

Description automatically generatedChart, line chart

Description automatically generated**

**Prediction Summary:**

* **SP500:** In the end, the best and chosen prediction model was the Gradient Boosting, with a learning rate of 0.1, 5 depth levels and 300 maximum number of iterations, yielding a desired performance R2 of 0.97.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

* **Pollution:** Overall the best prediction model was a Random Forests model with a R2=0.94. This model was achieved using a maximum of 75% of the total features available, a maximum depth of 20 and 75 estimators. The MLP achieved similar performance but fell a bit short to the random forest (see Annex).

# Forecasting

Both datasets were curated so that the each had only the time variable and the target variable. In **SP500**’s case the target is the Close Price, ***close***whereas in the **Pollution** dataset the target is the concentration of PM2.5, ***PM2.5.*** The same data treatments applied before were carried on to this phase.

## Basic Forecasting Models

## Simple Average

* + **SP500:** This naive model yielded a R2 of -0.03.

**Graphical user interface, application

Description automatically generated**

* + **Pollution:**
* ***Persistence***
  + **SP500:** The persistence model achieved a R2 of -0.88.

**Graphical user interface, application

Description automatically generated**

* + **Pollution:**

Shall be used to present the evaluation of the models achieved through Persistence and Simple Average. The results shall be used as baselines for comparing the following models results.

## KNN

Applied over the flattened dataset. Shall be used to present the results achieved through different similarity measures and parametrizations of KNN. The results shall be compared and explanations for them shall be presented. The justification for the chosen similarity measures shall be presented.

Shall be used to present the evaluation of the best model achieved.

* + **SP500:** The best results were achieved with uniform weight, Euclidean distance and k=19, despite all the distances yielding similar results. The number k was the most determinant factor to discriminate between performances. Nonetheless, the R2 was very poor: -0.22. (See Annex).

Graphical user interface, application

Description automatically generated

Overfitting seems to be a problem and ideally greater k values should be tested to determine the overfitting threshold.

**Chart, line chart

Description automatically generated**

* + **Pollution:**

## Regression Trees

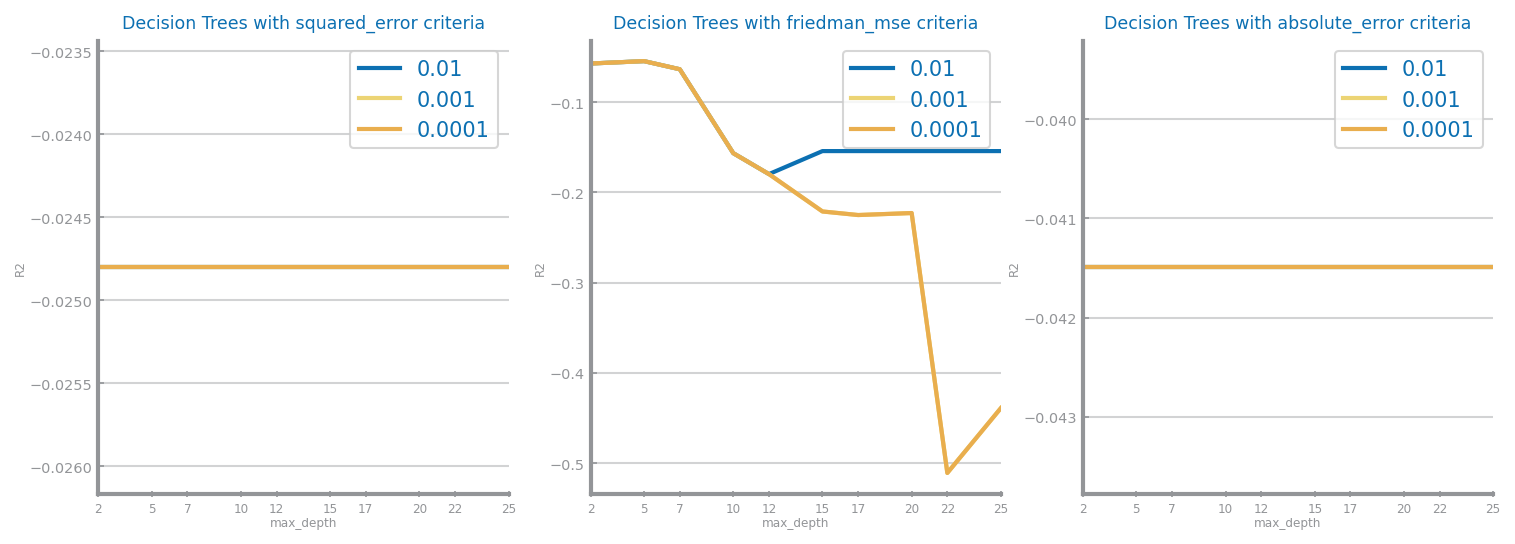
Applied over the flattened dataset. Shall be used to present the results achieved through different parametrizations for the train of regression trees. The results shall be compared and explanations for them shall be presented.

Shall be used to present the evaluation of the best model achieved.

Shall be used to present the best tree achieved and its succinct description.

* + **SP500:** The best results are achieved with the squared error criteria and minimum impurity decrease of 0.01 and depth=2 (since the graph is an horizontal line for all levels of impurity, all depth values tested have the same R2=0.02). The type of criteria used seems to be the most relevant variable in the models.

There are no apparent overfitting issues since the test R2 plot relative to max depth is a flat line (see Appendix).



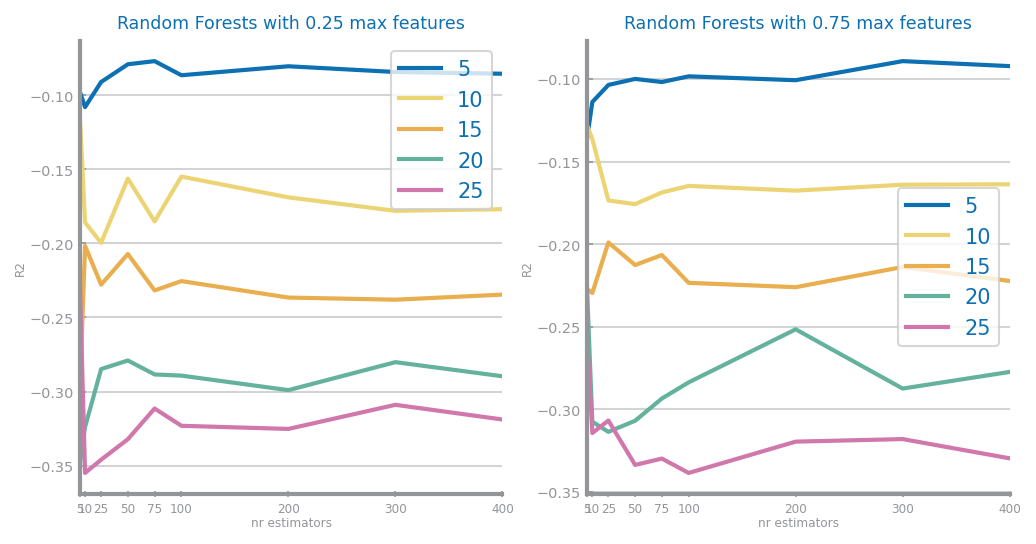
## Random Forests

Applied over the flattened dataset. Shall be used to present the results achieved through different parametrizations for the train of random forests. The results shall be compared and explanations for them shall be presented.

Shall be used to present the evaluation of the best model achieved.

May be used to present the most important variables in the model.

* + **SP500:** The best results with random forests (R2= -0.08) are achieved with 25% of the features, depth=5 and 75 estimators. The depth is clearly the variable that controls for the major differences in performance among the various tested scenarios and the lower the depth the better the results obtained. Nonetheless the results are very poor.



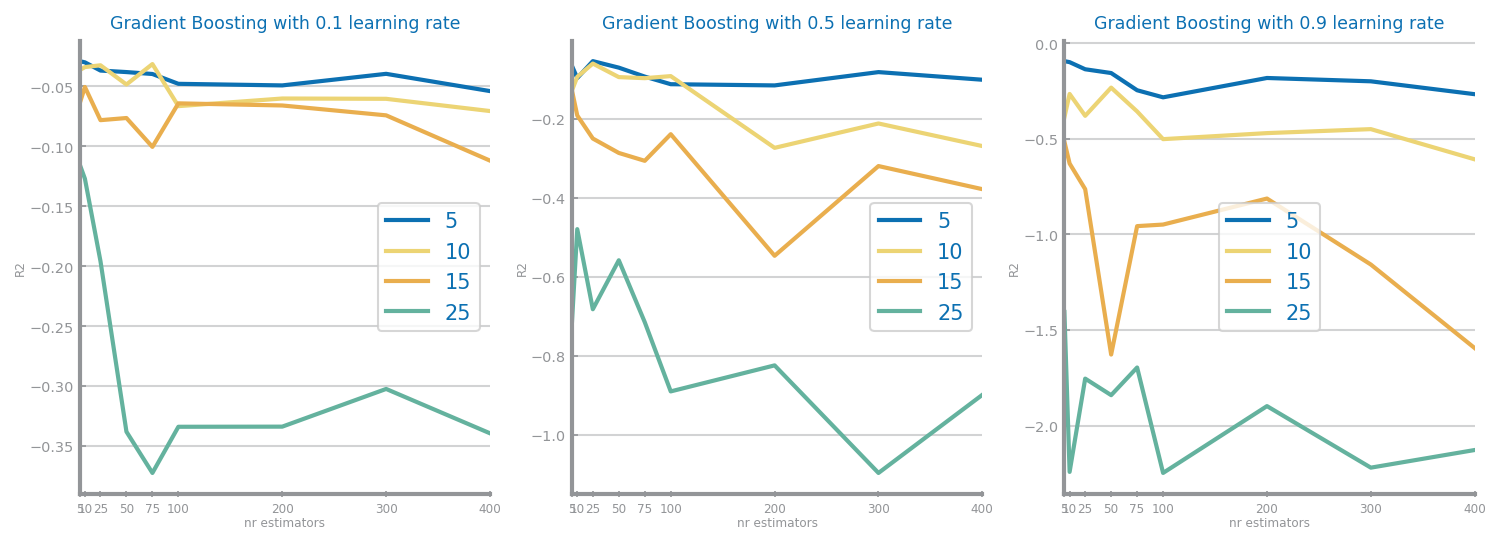
## Gradient Boosting

Applied over the flattened dataset. Shall be used to present the results achieved through different parametrizations for the train of gradient boosting. The results shall be compared and explanations for them shall be presented.

Shall be used to present the evaluation of the best model achieved.

May be used to present the most important variables in the model.

* + **SP500:** The best results are achieved with 0.1 learning rate, depth=5 and 5 estimators and the R2 value is -0.03. Again, the results obtained are very poor, however we can conclude that mainly the depth but also the learning rate and the number of estimators play an important role in the performance of the Gradient Boosting model for this dataset.



## Multi-Layer Perceptrons

Applied over the flattened dataset. Shall be used to present the results achieved through different parametrizations for the train of MLPs. The results shall be compared and explanations for them shall be presented.

Shall be used to present the evaluation of the best model achieved.

* + **SP500:** The best Multi-Layer Perceptron is attained with a constant learning rate of 0.3 and 1000 max iterations and it has a R2 of 0.00. The learning rates of 0.1 and 0.3 display a flat behavior and the bigger learning rate (0.9) is more sensible to the number of iterations.

Chart, line chart

Description automatically generated

The overfitting plot presents an interesting behavior that suggests that the model may enter in overfitting for more than 150 iterations.

Chart, line chart

Description automatically generated

## Rolling Mean

Applied over the target variable series. Shall be used to present the results achieved through different parametrizations (window size).The results shall be compared and explanations for them shall be presented.

Shall be used to present the evaluation of the best model achieved.

## ARIMA

Applied over the target variable series. Shall be used to present the results achieved through different parametrizations. The results shall be compared and explanations for them shall be presented.

Shall be used to present the evaluation of the best model achieved.

## LSTM

Applied over the target variable series. Shall be used to present the results achieved through different parametrizations. The results shall be compared and explanations for them shall be presented.

Shall be used to present the evaluation of the best model achieved.

# Critical Analysis

Shall be used to present a summary of the results achieved with the different modeling techniques, and the impact of the different preparation tasks on their performance.

A cross-analysis of the different models may also be presented, identifying the most relevant variables common to all of them (when possible) and the relation among the patterns identified within the different classifiers.

A critical assessment of the best models shall be presented, clearly stating if the models seem to be good enough for the problem at hand.

# Appendix (optional)

1. **Data Profiling**
   1. Data Granularity
      1. Graphical user interface, chart, line chart

         Description automatically generatedChart, line chart

         Description automatically generatedChart, line chart

         Description automatically generatedChart, line chart

         Description automatically generatedSP500:
      2. Polution:

Chart, line chart

Description automatically generated

Graphical user interface, chart

Description automatically generatedChart, line chart

Description automatically generated

* 1. Data Distribution
     1. Graphical user interface, chart, box and whisker chart

        Description automatically generatedA screenshot of a computer

        Description automatically generated with low confidenceSP500:
     2. Polution:

A picture containing text

Description automatically generatedChart, box and whisker chart

Description automatically generated

* 1. Data Stationarity

1. **Data Preparation:**
   1. Base Model (No preparation):
      1. SP500:

Best results with lr\_type=constant, learning rate=0.1 and 500 max iter ==> measure=0.67

{'RMSE': [0.014520273526434618, 0.04971838570700506], 'MAE': [0.01001534102240972, 0.044681847718953734]} {'R2': [0.99976649087693, 0.6708140844631139]}

**Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated**

Chart, bar chart

Description automatically generated

1. Pollution:

Best results with lr\_type=constant, learning rate=0.1 and 150 max iter ==> measure=0.88

{'RMSE': [0.34442850807980624, 0.35360555691886286], 'MAE': [0.21535954378644584, 0.21541793247139063]} {'R2': [0.8781625358159028, 0.8816355187315851]}

Graphical user interface, text, application, email

Description automatically generatedChart, bar chart

Description automatically generatedChart, line chart

Description automatically generated

* 1. Aggregation
     1. SP500:
        1. Monthly:

Best results with lr\_type=constant, learning rate=0.1 and 250 max iter ==> measure=0.54

Chart, line chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, line chart

Description automatically generated{'RMSE': [0.01681080258217593, 0.059002749698116974], 'MAE': [0.011501606380514715, 0.05214846547324733]} {'R2': [0.9996870095013672, 0.5363911584219161]}

* + - 1. Quarterly:

Best results with lr\_type=constant, learning rate=0.1 and 500 max iter ==> measure=0.63

{'RMSE': [0.012313022426090477, 0.05243286551526763], 'MAE': [0.007564542024854479, 0.04545778641851766]} {'R2': [0.9998320872792031, 0.6338876382815802]}

Graphical user interface, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, bar chart

Description automatically generated

* + - 1. Yearly:

Best results with lr\_type=constant, learning rate=0.1 and 1000 max iter ==> measure=0.25

Chart, line chart

Description automatically generated{'RMSE': [0.017007391186966845, 0.07496293681611287], 'MAE': [0.012473537919595571, 0.06785574243666646]} {'R2': [0.9996796463622407, 0.25165753184910844]}

Graphical user interface, chart, line chart

Description automatically generatedChart, bar chart

Description automatically generated

* + 1. Polution:
       1. Daily:

Best results with lr\_type=constant, learning rate=0.1 and 150 max iter ==> measure=0.93

{'RMSE': [0.18164084318343426, 0.2438110583628574], 'MAE': [0.1338178619394738, 0.1656881999849065]} {'R2': [0.9527478928891546, 0.9260697332475243]}

Chart, line chart

Description automatically generatedGraphical user interface, application

Description automatically generatedChart, bar chart

Description automatically generated

* + - 1. Weekly:

Best results with lr\_type=constant, learning rate=0.1 and 150 max iter ==> measure=0.90

{'RMSE': [0.09181148471822388, 0.1805485169352684], 'MAE': [0.06544773351969241, 0.13139791989279404]} {'R2': [0.9633288042642258, 0.8998901701916284]}

Chart, bar chart

Description automatically generatedGraphical user interface, chart, line chart

Description automatically generated

* + - 1. Monthly:

Best results with lr\_type=constant, learning rate=0.1 and 500 max iter ==> measure=0.85

{'RMSE': [0.06490519052714037, 0.13266717426539357], 'MAE': [0.029877083646306563, 0.1004488057218384]} {'R2': [0.9536057440115465, 0.8547668862562503]}

Chart, bar chart

Description automatically generatedChart, line chart

Description automatically generated

* + - 1. Quarterly

Best results with lr\_type=constant, learning rate=0.1 and 500 max iter ==> measure=0.76

{'RMSE': [0.0191628833624158, 0.12271651043601257], 'MAE': [0.015636022338813404, 0.09988463812298458]} {'R2': [0.992860417367402, 0.7597220257903694]}

Chart, bar chart

Description automatically generatedChart, line chart

Description automatically generated

* 1. **Differentiation:**
     1. SP500:

Best results with lr\_type=constant, learning rate=0.1 and 150 max iter ==> measure=0.73

{'RMSE': [0.013052253628387559, 0.045261929271791554], 'MAE': [0.007904182907866864, 0.04222166011565079]} {'R2': [0.9998113202611298, 0.7271818056601179]}

Chart, bar chart

Description automatically generatedChart, line chart

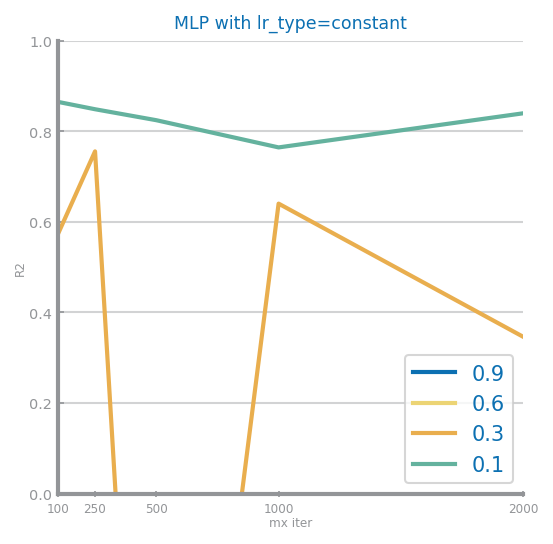
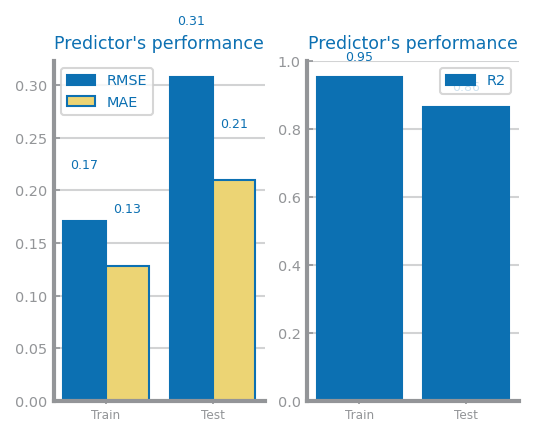
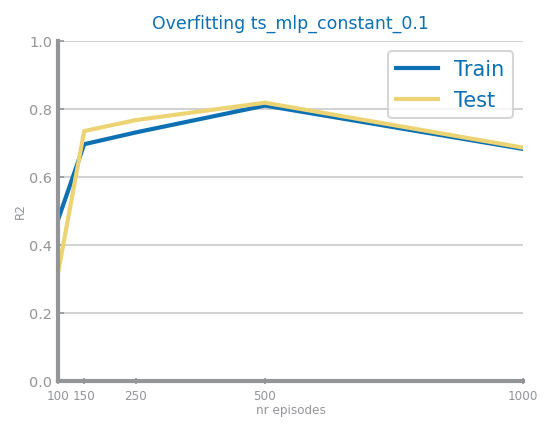
Description automatically generatedChart, line chart

Description automatically generated

* + 1. Polution:

Best results with lr\_type=constant, learning rate=0.1 and 100 max iter ==> measure=0.86

{'RMSE': [0.1705630783827969, 0.3079593054993992], 'MAE': [0.1285592822836654, 0.2103892113023254]} {'R2': [0.9532650266528979, 0.8644434889794042]}



* 1. Smoothing:
     1. SP500:
        1. 10 bins:

Best results with lr\_type=constant, learning rate=0.1 and 250 max iter ==> measure=0.21

{'RMSE': [0.01143273393754861, 0.07715624035403443], 'MAE': [0.007624200492591373, 0.0674344455684923]} {'R2': [0.9998552380410883, 0.20722614045028676]}

Chart, line chart

Description automatically generated

Chart, bar chart

Description automatically generatedGraphical user interface, line chart

Description automatically generated

* + - 1. 100 bins:

Best results with lr\_type=constant, learning rate=0.1 and 150 max iter ==> measure=-7.65

{'RMSE': [0.01116891266435983, 0.254906765091503], 'MAE': [0.006184628193383649, 0.21549451991467802]} {'R2': [0.9998618419967659, -7.653064282638933]}

Chart

Description automatically generatedGraphical user interface, application

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

* + 1. SP500:
       1. 10 bins:

Best results with lr\_type=constant, learning rate=0.1 and 500 max iter ==> measure=0.89

{'RMSE': [0.08648274844565958, 0.15806624305748257], 'MAE': [0.06760918044012117, 0.11971285879298715]} {'R2': [0.9548238101224583, 0.8929019358113174]}

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, bar chart

Description automatically generated

* + - 1. 100 bins:

Best results with lr\_type=constant, learning rate=0.1 and 100 max iter ==> measure=0.92

{'RMSE': [0.025955380569682673, 0.06603372369714074], 'MAE': [0.019007887974598688, 0.05163822106932313]} {'R2': [0.9796317653428493, 0.9157966920240153]}

Graphical user interface, chart, line chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, line chart

Description automatically generated

1. **Prediction**
   1. KNN
      1. SP500

Best results achieved with uniform weight, dist=manhattan and K=19 ==> measure=0.92

{'RMSE': [0.011729608067119276, 0.0006699246825479816], 'MAE': [0.00588246419000329, 0.0003085569861825963]} {'R2': [0.6650502029479886, 0.9232386244150398]}

Chart, bar chart

Description automatically generatedGraphical user interface, application

Description automatically generated

Chart, line chart

Description automatically generated

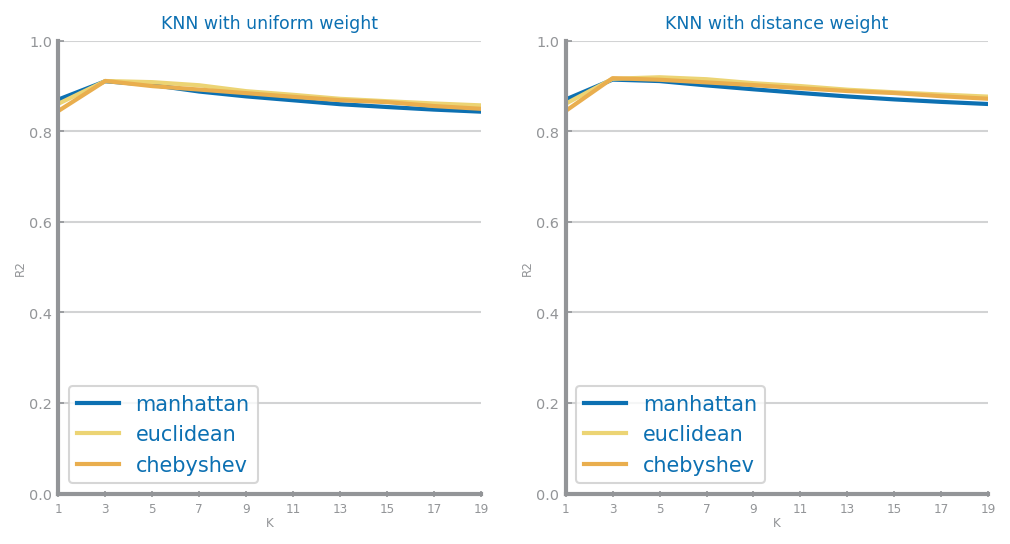
* + 1. Pollution:

Best results achieved with distance weight, dist=euclidean and K=5 ==> measure=0.92

{'RMSE': [0.0, 0.25522879620328137], 'MAE': [0.0, 0.17286259468777548]} {'R2': [1.0, 0.9189832498872438]}

Graphical user interface

Description automatically generatedChart, bar chart

Description automatically generated

* 1. Decision Tree
     1. SP500

Best results achieved with friedman\_mse criteria, min\_impurity\_decrease=0.0001 and depth=15 ==> measure=0.88

A picture containing text, screenshot, computer

Description automatically generated{'RMSE': [0.005780648707558898, 0.0008348492330895619], 'MAE': [0.0035690783247934736, 0.0005564313981611022]} {'R2': [0.9186484422992237, 0.8807915899318134]}

Chart, bar chart

Description automatically generatedChart, line chart

Description automatically generated

* + 1. Pollution

Best results achieved with mae criteria, min\_impurity\_decrease=0.001 and depth=5 ==> measure=0.92

{'RMSE': [0.25463455312242284, 0.25587096754665506], 'MAE': [0.17030919861726834, 0.1674446990306012]} {'R2': [0.9071136296491173, 0.9185750507566041]}

Chart, line chart

Description automatically generatedChart, bar chart

Description automatically generatedGraphical user interface, application, line chart

Description automatically generated

* 1. Random Forest
     1. SP500

Best results achieved with 0.75 max features, depth=15 and nr estimators=200 ==> measure=0.93

{'RMSE': [0.0061398683079872726, 0.0006527416875743962], 'MAE': [0.0038353377322598576, 0.0002979108037534696]} {'R2': [0.9082236398280575, 0.9271258519735398]}

A screenshot of a computer

Description automatically generated with medium confidence

Chart, line chart

Description automatically generated

Chart, bar chart

Description automatically generated

* + 1. Pollution

Best results achieved with 0.75 max features, depth=20 and nr estimators=75 ==> measure=0.94

{'RMSE': [0.09834681864636477, 0.2218534404948356], 'MAE': [0.06308255882371976, 0.14630060215617335]} {'R2': [0.9861439975098034, 0.9387864156836632]}

Line chart

Description automatically generatedChart, bar chart

Description automatically generated

Graphical user interface, line chart

Description automatically generated

* 1. Gradient Boosting
     1. SP500

Best results achieved with 0.1 learning rate, depth=5 and nr estimators=300 ==> measure=0.97

A screenshot of a computer

Description automatically generated with low confidence{'RMSE': [0.00931350150637047, 0.0004368358272862313], 'MAE': [0.004742291088002005, 0.0001315657402432664]} {'R2': [0.7888268512766593, 0.9673617179922914]}

Graphical user interface, text, application

Description automatically generatedChart, bar chart

Description automatically generated

* + 1. Pollution

Best results achieved with 0.1 learning rate, depth=25 and nr estimators=400 ==> measure=0.93

{'RMSE': [0.003249035546555605, 0.23866249735620274], 'MAE': [0.0003693166417590735, 0.15019334284754174]} {'R2': [0.9999848774075979, 0.929159138294535]}

A picture containing text, screenshot, electronics, display

Description automatically generatedGraphical user interface, application

Description automatically generatedChart

Description automatically generated

* 1. Multi Layer Perceptrons
     1. SP500

Same results as for the differentiation treatment since the performance comparison of inplemeting differentiation as done using MLP.

* + 1. SP500

Best results with lr\_type=invscaling, learning rate=0.1 and 150 max iter ==> measure=0.94

{'RMSE': [0.20007712082995968, 0.22736510290679632], 'MAE': [0.14086854085231007, 0.1578270444686336]} {'R2': [0.9426528051139416, 0.9357070890495207]}

Chart, line chart

Description automatically generatedChart, bar chart

Description automatically generatedA screenshot of a computer

Description automatically generated with low confidence

1. **Forecasting**
   1. Basic Forecasting Models
2. SP500

{'SimpleAvg': -0.026344010292771003} {'RMSE': [0.01959696144430312, 0.002259408380117152], 'MAE': [0.01092451524786873, 0.0012918607803891423]} {'R2': [1.1102230246251565e-16, -0.026344010292771003]}

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

-0.0011820412087407695 {'SimpleAvg': -0.026344010292771003, 'Persistence': -0.8816445844310998} {'RMSE': [0.02936230175296468, 0.0030592647073911967], 'MAE': [0.01621196951165478, 0.0017679489719742995]} {'R2': [-1.2449295610438025, -0.8816445844310998]}

Graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated

* 1. KNN

1. SP500

Best results achieved with uniform weight, dist=euclidean and K=19 ==> measure=-0.22

{'RMSE': [0.01916512376675719, 0.002671210594220306], 'MAE': [0.01107443290235911, 0.0017028549951085665]} {'R2': [0.09630734830301013, -0.22052728992730697]}

Graphical user interface, application

Description automatically generatedChart, line chart

Description automatically generated

Graphical user interface, application

Description automatically generated

1. Pollution:
   1. Decision Tree
2. SP500

Best results achieved with squared\_error criteria, min\_impurity\_decrease=0.01 and depth=2 ==> measure=-0.02

{'RMSE': [0.020160497863311654, 0.002447675220949089], 'MAE': [0.011506337597520667, 0.001475653637732575]} {'R2': [-2.220446049250313e-16, -0.02479929955487714]}

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application, line chart

Description automatically generated

1. Pollution:
   1. Random Forest
2. SP500

Best results achieved with 0.25 max features, depth=5 and nr estimators=75 ==> measure=-0.08

{'RMSE': [0.01854986620446213, 0.002509293308527831], 'MAE': [0.011227278255911638, 0.001568713256422546]} {'R2': [0.1533984536352333, -0.07704560849467934]}

Graphical user interface, chart, line chart

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated

1. Pollution:
   1. Gradient Boosting
2. SP500

Best results achieved with 0.1 learning rate, depth=5 and nr estimators=5 ==> measure=-0.03

{'RMSE': [0.0196109966002762, 0.00245261038768371], 'MAE': [0.011271672736026379, 0.0014910474065299635]} {'R2': [0.053769760224721574, -0.028936003689881584]}

A screenshot of a computer

Description automatically generated with low confidenceGraphical user interface, text, application, table

Description automatically generatedGraphical user interface, application

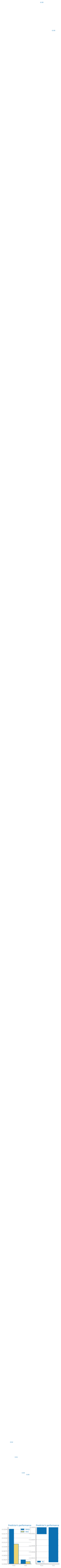
Description automatically generated

1. Pollution
   1. Multi Layer Perceptrons
2. SP500

Best results with lr\_type=constant, learning rate=0.3 and 1000 max iter ==> measure=-0.00

{'RMSE': [0.020162717669127733, 0.002419253887082559], 'MAE': [0.011515352297584757, 0.0014328921120815086]} {'R2': [-0.00022022551614275798, -0.0011384293059213402]}

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

1. Pollution
   1. Rolling Mean
2. SP500

{'SimpleAvg': -0.026344010292771003, 'Persistence': -0.8816445844310998, 'RollingMean': 0.31683315010669466} {'RMSE': [0.016047429942985584, 0.0018433663331111976], 'MAE': [0.009115164673263135, 0.0010486740475631675]} {'R2': [0.32944639694091504, 0.31683315010669466]}

Graphical user interface, application

Description automatically generatedA picture containing text, screenshot

Description automatically generated

* 1. ARIMA
     1. SP500

Best results achieved with (p,d,q)=(5, 2, 1) ==> measure=0.80

R2=0.804023673862601 {'RMSE': [0.031801561075820356, 0.4091014725329128], 'MAE': [0.010314336241989639, 0.24097037956667652]} {'R2': [0.9955204838634467, 0.804023673862601]}

Chart, bar chart

Description automatically generatedChart, line chart

Description automatically generated

* + 1. Pollution
  1. LSTM
     1. SP500
     2. Pollution