|  |  |  |  |
| --- | --- | --- | --- |
|  | | **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.
* **Polution:**

## 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).
* **Polution:**

## 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.
* **Polution:**

## 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.
* **Polution:**

# 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 utilized is the . As a base model for performance comparison, we ran the MLP without any of the preparations mentioned, yielding a R2 = 0.67.

## 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.
* **Polution:**

## 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

* **Polution:**

## 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).
* **Polution:**

# 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

* **Polution:**

## 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

* **Polution:**

## 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

* **Polution:**

## 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.

## 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 phenomenoma 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.

**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

* Polution:

# Forecasting

Shall be used to summarize the preparation applied to perform the forecasting task. It shall be applied just using the **target variable**.

## Basic Forecasting Models

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.

## 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.

## 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.

## 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.

## 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.

## 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:
   2. 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:
   3. Data Stationarity
2. **Data Preparation:**
   1. Base Model (No preparation):

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. 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. **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:
  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. **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. 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. 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. 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. Multi Layer Perceptrons
     1. SP500

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

1. **Forecasting**
   1. Basic Forecasting Models
   2. KNN
   3. Decision Tree
   4. Random Forest
   5. Gradient Boosting
   6. Multi Layer Perceptrons
   7. Rolling Mean
   8. ARIMA
   9. LSTM