Case Study for Data Analyst/Scientist Internship at ID&A TECH

PREDICTION OF THE PERFORMANCE OF THE MASI INDEX

EL-FAILALI-LBALGHITI Mohamed
Data Scientist - ENSA Marrakech
12/17/2023

Method

Transformation of data

At this step, min-max normalization is used to execute data type conversions, scaling, and normalization. This process ensures data consistency and uniformity. To enact these transformations, we used the following formula:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

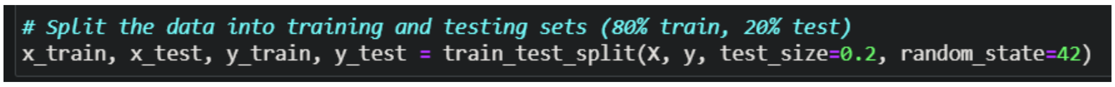
Test and training data

The dataset is divided into two sets, namely the training and test data, following an 80:20 ratio.

The training data, consisting of 3963 observations, is exclusively utilized for training predictive models.

The machine learning algorithm examines patterns and learns from diverse observations within this set.

Conversely, the test data, comprising 991 observations, is employed to assess the accuracy of the trained model and generate predictions. By preserving the actual data for the forecast period, we can evaluate the precision of our predictions and compare them to the real values



```
# Split the data into training and testing sets (80% train, 20% test)
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Test and training data

To develop the prediction model at this level, we used the ML ANN algorithms and provided them with the features of the training data. ANN consist of many layers that connect input to output.

Choosing the model

Feature selection

Aims to eliminate extraneous variables, reduce training time, and prevent overfitting. Recursive feature selection determines the relative relevance depending on prediction accuracy. Several studies employ feature selection algorithms to forecast changes in stock prices

Evaluating the model

In this study, MAE us used to evaluate the models performance. These metrics provide a comprehensive assessment of the model's predictive accuracy and robustness. The equations is:

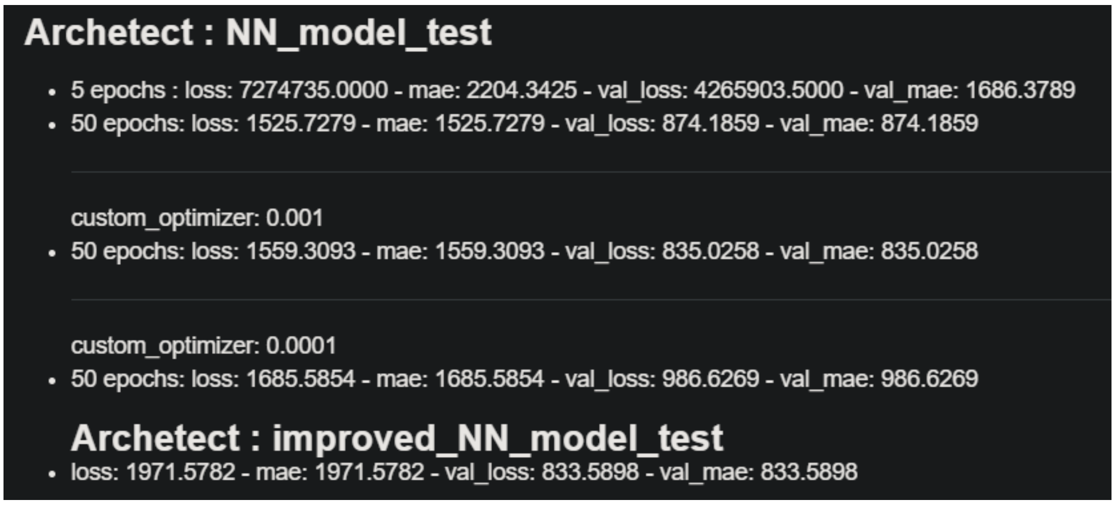
$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$$

Result

For this problem we used Simple FNN, LSTM and Random forest

1. FNN:

Determining the optimal number of hidden layers for a deep learning model poses a significant challenge due to the absence of a definitive rule, so in this case I test different architectures



Archetect : NN_model_test

- 5 epochs : loss: 7274735.0000 - mae: 2204.3425 - val_loss: 4265903.5000 - val_mae: 1686.3789
- 50 epochs: loss: 1525.7279 - mae: 1525.7279 - val_loss: 874.1859 - val_mae: 874.1859

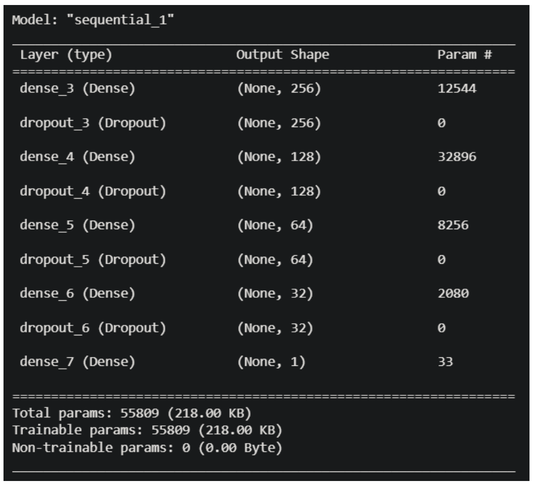
custom_optimizer: 0.001
- 50 epochs: loss: 1559.3093 - mae: 1559.3093 - val_loss: 835.0258 - val_mae: 835.0258

custom_optimizer: 0.0001
- 50 epochs: loss: 1685.5854 - mae: 1685.5854 - val_loss: 986.6269 - val_mae: 986.6269

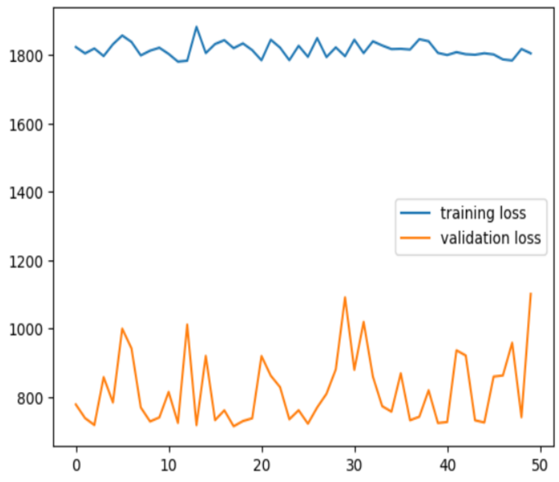
Archetect : improved_NN_model_test
- loss: 1971.5782 - mae: 1971.5782 - val_loss: 833.5898 - val_mae: 833.5898

We find that the improved FNN give us the best accuracy:

Figure in the left shows our proposed architecture for model, with an input layer of 256 nodes, 3 hidden layers, and an output layer.

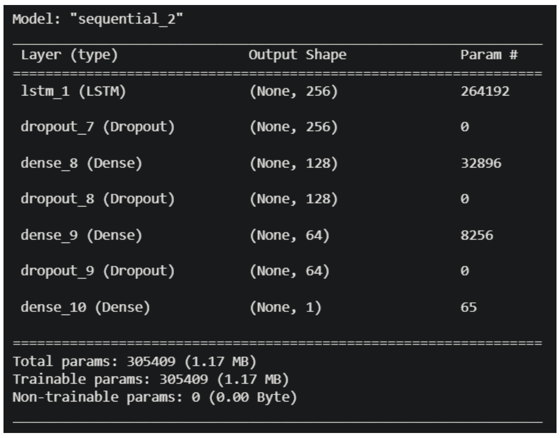


```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 256)               12544

 dropout_3 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 128)               32896

 dropout_4 (Dropout)         (None, 128)               0

 dense_5 (Dense)             (None, 64)                8256

 dropout_5 (Dropout)         (None, 64)                0

 dense_6 (Dense)             (None, 32)                2080

 dropout_6 (Dropout)         (None, 32)                0

 dense_7 (Dense)             (None, 1)                 33

=================================================================
Total params: 55809 (218.00 KB)
Trainable params: 55809 (218.00 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

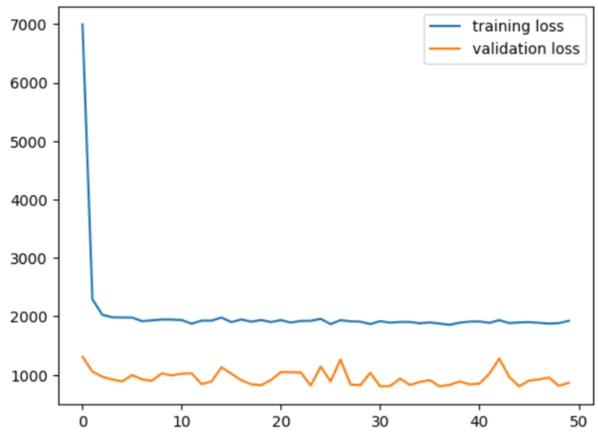


FNN Model Architecture and training/validation loss

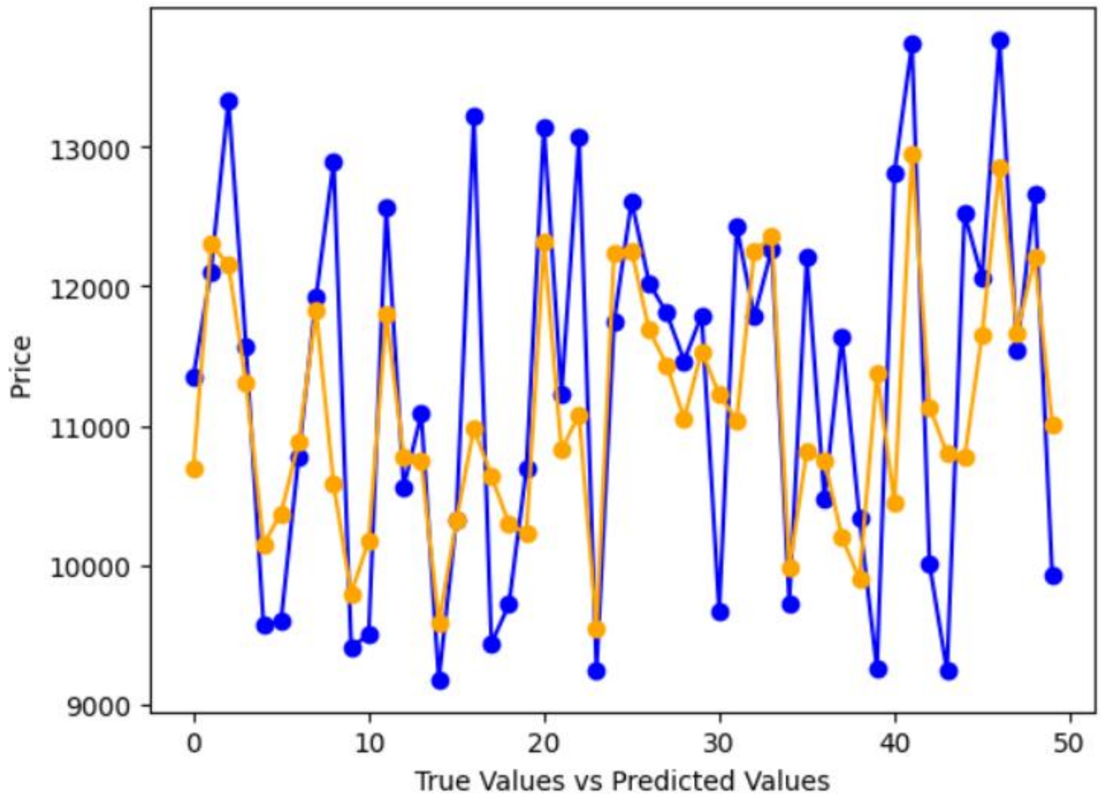
2. LSTM:



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_1 (LSTM)               (None, 256)               264192

 dropout_7 (Dropout)         (None, 256)               0

 dense_8 (Dense)             (None, 128)               32896

 dropout_8 (Dropout)         (None, 128)               0

 dense_9 (Dense)             (None, 64)                8256

 dropout_9 (Dropout)         (None, 64)                0

 dense_10 (Dense)            (None, 1)                 65

=================================================================
Total params: 305409 (1.17 MB)
Trainable params: 305409 (1.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



LSTM Model Architecture and training/validation loss

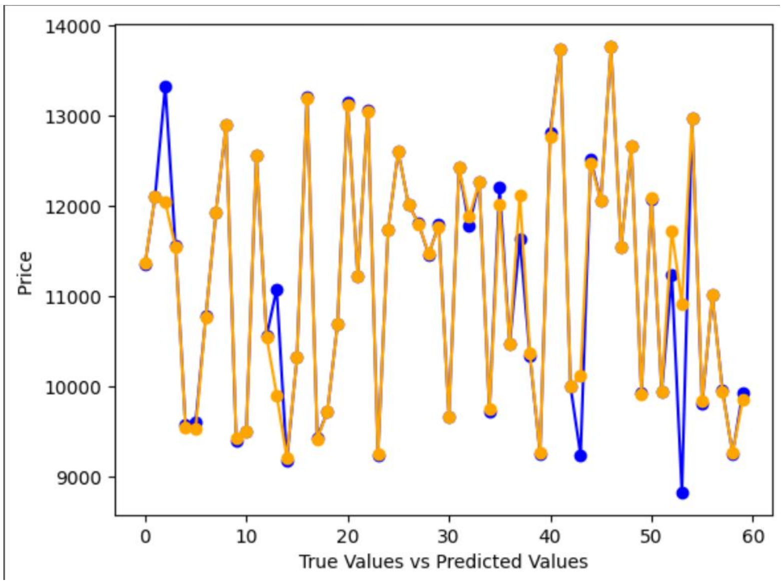


Upon completing the training phase and applying our model for forecasting purposes, we obtained remarkable results regarding MAE. The calculated MAE, indicates the extent to which the predicted values deviate from the expected values.

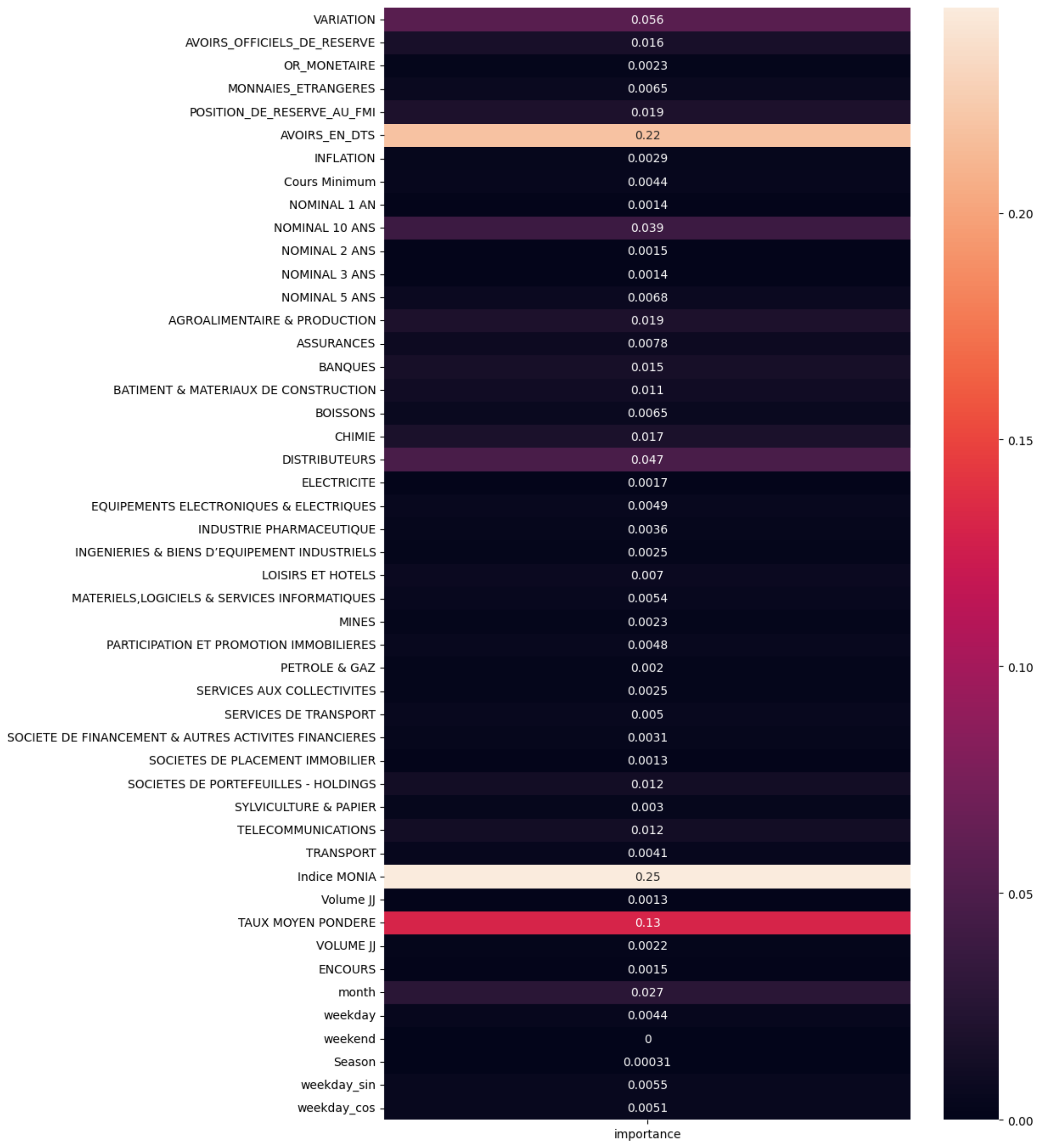
These evaluation metrics demonstrate our model's good accuracy and precision in capturing and predicting the underlying models within the data. As illustrated in Figure above the graph visually showcases the close alignment between the predicted and actual values,

3. Random Forest:

The Mean Absolute Error (MAE) on the validation set is a metric used to assess the performance of the model in predicting the target variable. In this context, a MAE of 92.337 indicates that, on average, the model's predictions deviate by approximately 92.337 units from the actual values in the validation set.



The feature importance analysis provides valuable insights into the factors influencing the model predictions. In this context, the Random Forest model has identified key features contributing to its decision-making process. Notably, the most influential feature is "Indice MONIA," accounting for a substantial portion of the predictive power. Other significant contributors include "AVOIRS_EN_DTS," "TAUX_MOYEN_PONDERE," and "VARIATION," reflecting the importance of economic indicators and financial metrics in shaping the model's output. Additionally, industry sectors such as "DISTRIBUTEURS" and "BANQUES" play a role, underlining the impact of market dynamics within specific sectors. The temporal aspects, represented by "month," also contribute significantly, emphasizing the relevance of time-related patterns in the dataset. The absence of importance for certain features like "weekend" and "Season" suggests that these variables may have limited influence on the model predictions. Overall, this feature importance analysis provides a comprehensive understanding of the model's decision criteria, aiding in the interpretation and refinement of the predictive model.



Features importances

NEXT WORK:

- Manual selection, in such cases, tends to be impractical and subjective. However, a promising solution emerges through automated techniques like the Keras Tuner library.
  This powerful tool enables efficient search space exploration, facilitating the identification of the most suitable model configuration through a series of systematic trials. The Keras Tuner empowers researchers to define the architecture of hidden layers by specifying the number of nodes in each layer.
- Refine the model by eliminating less significant features and subsequently engaging in fine-tuning for enhanced performance.