**DEEP LEARNING PROJECT REPORT**

**TRACK B OCTOBER 2023**

**INTRODUCTION - STOCK PRICES**

Stock price forecasting is a fascinating and complex challenge that intersects the fields of finance and machine learning. This field requires the ability to accurately anticipate future stock prices fluctuations, especially Open, High, Low, Close values based on the same historical data of the financial markets and on a number of key indicators, such as Trading Volume, for example.

During this report, we will explore in detail the foundations of Deep Learning, delving into the selection and pre-processing of the datasets used. In addition, we will carefully analyze the implementation of various predictive models, with the aim of revealing the results obtained and commenting on them.

**UTILS**

* Google Colab: Google Colaboratory, or Colab for short, is a cloud-based platform provided by Google for running Python code in Jupyter notebooks. It offers free access to GPU and TPU resources for machine learning and data analysis tasks.
* TensorFlow: TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and community resources for building and deploying machine learning models.
* Python: Python is a high-level, general-purpose programming language known for its simplicity and readability. It is widely used in data science, machine learning, web development, and various other fields.
* Keras: Keras is an open-source high-level neural networks API written in Python. It is designed to be user-friendly, modular, and extensible, making it a popular choice for building neural networks and deep learning models. Keras is integrated with TensorFlow and is often used as a high-level API for TensorFlow.

**DATASETS**

The datasets used were those of Yahoo Finance and concern companies: Amazon, Apple, Microsoft and Google and all these have the same date range ranging from 21-August 2019 to 18-08-2023, so 4 years of values day by day.

These datasets consist of several values:

Image containing text, screenshot, Font, number

Auto-generated description

We have:

* Dates that is already set in daily mode (day by day) by default
* Open: This column indicates the stock’s opening price at the beginning of the trading day. It is the value at which the security was traded at the time of the market opening.
* High: This column indicates the maximum price reached by the stock during the trading day. It is the highest value recorded during the day.
* Low: This column represents the minimum price touched by the stock during the same trading day. It is the lowest value recorded during the day.

Close: This column represents the closing price of the stock at the end of the trading day. It is the value at which the security was trading at the time of the market close.

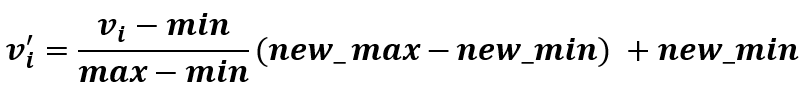
* Volume (Trading Volume): This column indicates the total amount of shares in the security that were traded during the trading day. Volume is an important indicator of the liquidity of the stock and its popularity among investors.
* Adj Close: This column represents the closing price of the stock, but has been "adjusted" for any events such as dividends, stock splits, or other company changes that may affect the price. Adjustment is important to accurately calculate returns over time.

**DATA PRE PROCESS**

The following techniques were used for data pre-processing:

* str\_to\_datetime: This is a function defined to convert a date string in the format "YYYY-MM-DD" into a datetime object. This is useful for manipulating dates more easily.
* Loading data from CSV files.
* Selection of columns of interest: A subset of the columns from the DataFrame has been selected which includes: 'Date', 'Close', 'Open', 'High', 'Low' and 'Volume'. These columns contain the most relevant information for stock price analysis, in fact the Adj Close column is discarded for redundancy.
* Date conversion: The 'Date' column is converted from strings to datetime objects using the str\_to\_datetime function. It is then set as the index of the DataFrame to make it easier to manipulate temporal data.
* Removing of missing or null data: Rows in the DataFrame that contain missing or null data are deleted using the dropna method.
* Data normalization: The data in the 'Close', 'Open', 'High', 'Low', and 'Volume' columns are normalized using the Min-Max technique, so that all values are in the range [0, 1] = [newmin, newmax].

This normalization is important to ensure that different characteristics have the same weight when training the model.



* Creating inputs (X) and targets (y): Normalized data is divided into input (X) and target (y). In this case, both X and y contain the same columns: 'Close', 'Open', 'High', 'Low' and 'Volume' since we are making a multi-input, multi-output prediction.
* Set size calculation: The size of the training set (train\_size) and the size of the validation set (val\_size) is calculated, where 20% of the training set is used as the validation set.
* Split data: Data is divided into training, validation, and test sets based on previously calculated dimensions.
* Adding a dummy time dimension for features: Three sets of 3D inputs (X\_train\_3d, X\_val\_3d, X\_test\_3d) are created by adding a timestep of 1. This is done because the models we use require a specific size to the input data which is: (Batch Size, TimeStep, Feature Dimension).
  + Batch Size: is the number of samples that is processed simultaneously
  + TimeStep: is the dimension we added and is equal to 1 because the data represents a time series of daily prices and therefore it is important to make the model consider one data point at a time
  + Feature Dimension: The number of features chosen: 5.
* Add a time dimension for targets as well: Similarly, targets are transformed into 3D tensors (y\_train\_3d, y\_val\_3d, y\_test\_3d) by adding a time dimension of length 1.

**MODELS**

The models used for the prediction of time series in this project are 4:

* **Bi-lstm model**: with an Multi-head Attention module it is designed to process time series data with the aim of predicting the next observations in the series. Bidirectional LSTMs and attention layer allow the model to capture complex time dependencies in data, while dropouts and regularizations help control overfitting.
* **Simplified Transformer**: Designed to process data with 5 characteristics using a multi-head attention module and dense layers for learning complex time series.
* **Hybrid Model (mixture model):** combines the predictions of previous models using the average operation, i.e. the predictions generated by the two models are put together by calculating the average of the expected values. This is done to exploit the different temporal relationship capture capabilities of the two approaches.
* **Hybrid Model (with Weighted Attention):** takes time series data in input and uses previous models (Bi-lstm and Simplified Transformer) as two sub-models to generate predictions based on both approaches. It calculates attention weights using the output of the Bi-LSTM model and applying them to the outputs of the two models. All this allows the model to give greater importance to one of the two forecast sources based on the calculated attention weights, then the weighted outputs are combined and further processed through dense layers to generate the final output with 5 forecasts. This hybrid approach leverages the strengths of both architectures and allows the model to better adapt to time series data.

**MODEL BI-LSTM**

A Bidirectional Long Short-Term Memory (LSTM) network is a variant of the LSTM (Long Short-Term Memory) network and is bi-didirectional because it can capture information from both past and future in a sequence of data. To understand how a Bi-LSTM network works and how it differs from a regular LSTM, it is useful to first examine how a standard LSTM works.

**LSTM (Long Short-Term Memory):**

An LSTM network is an advanced recurrent neural network (RNN) that is designed to:

* Solving the problem of long-term dependencies
* Manage problems where the timeline of data is important, in fact they are equipped with an internal memory that can store long-term information and allow the passage of information from one time state to another.

There are 4 gates that control the flow of info in memory cells:

Image that contains diagram

Auto-generated description

First State: FORGET in which we want the irrelevant parts of the information of the previous state not to be kept and therefore to be forgotten.

* Use the output (ht-1) and input (xt) of the previous cell that are passed to a Sigmoid function that will return an ft value with 2 possibilities:
  + Value 0: "forget completely"
  + Value 1: "Keep completely".

Second State: STORE (Input) where LSTMs store new relevant information in the state of the cell.

* First, we pass the previous Hidden State (ht-1) and the current input (xt) to a Sigmoid function that decides which values will be updated, turning them into values between 0 (unimportant) and 1 (important). 🡪 son-in-law it.
* At the same time, the Hidden State (ht-1) and the current input (xt) are also passed into the tanh function, to normalize the values between -1 and 1🡪 Network Regularization.
* Then multiply the output of the tanh layer with the output of the sigmoid layer (en), and the output of the sigmoid will decide what information is important to maintain of the tanh output.

Third State: UPDATE: it is a gate that updates with selection the values of the state of the cells

* First, the previous cell state ct-1 is multiplied by element-wise by the forget vector. 🡪 CT-1 \* FT
  + The Forget vector has the ability to delete cell state values if it is multiplied by values close to 0.
* Then you take the output from the input gate: candidate \* en and perform a point-wise addition that updates (update) the state of the cell with the new values that the neural network deems relevant, in this way, you get the new state of the cell 🡪 then ct = ct-1 \* ft + candidate \* it.

Fourth State: OUTPUT Gate that controls what information is transmitted to the next step.

* Sigmoid Layer: decides which parts of the state to issue and which not. 🡪 Ot
* Tanh Layer: normalizes values between 1 and 1 of the current ct cell state
* ht = ot ∗ ct: New Hidden state (filtered output version of cell state ct).
* The new Cell state and the new Hidden state are then brought back to the next time step, but the hidden state is also taken as output y (forecast), in fact remember that it contains information on previous inputs, but is also used to make predictions.

Image containing text, clock

Auto-generated description

Image containing diagram, screenshot, text, plane

Auto-generated description

**Bi-LSTM (Bidirectional Long Short-Term Memory):**

A Bi-LSTM network works similarly to a regular LSTM, but has some key differences:

* Bidirectional: A Bi-LSTM network has two sets of LSTMs, one to capture information from the past (backward LSTM) and one to capture information from the future (forward LSTM). This means that it can consider both past and future context for each data point.
* Concatenated Output: The outputs of the two LSTMs (forward and backward) are concatenated or combined via a Trigger function to create the final output. This allows the model to have a more complete view of the temporal context.
* Greater Ability to Capture Time Dependencies: A Bi-LSTM network is able to capture longer-term relationships in timelines than a standard LSTM.

Image containing text, diagram, font, screenshot

Auto-generated description

The Bi-lstm model used in the project benefits from the use of bidirectional LSTMs and a multi-head attention module to capture complex time dependencies in the data in order to focus on the most relevant parts of the sequence.

The following are also introduced:

* Regularization that helps to improve the generalization of the model,
* A learning rate reduction callback that helps optimize training.

Its characteristics, specifically, are:

* Number of Features: The input dataset has 5 features
* Activation Function: The "ReLU" (Rectified Linear Unit) activation function is used between various layers of the model to introduce nonlinearity into the data.
* Two layers of Bi-LSTM to capture information from both the past and the future. These layers have 512 neurons in the first and 256 neurons in the second, and are set to return complete sequences (return\_sequences=True).
* Two layers of Dropout both with a 20% chance to reduce overfitting.
* Multi-head Attention Module, which is used to assign weights to the outputs of the Bi-LSTM according to their relevance to the task. This module allows the model to focus on specific parts of the timeline.
* Dense layers: After the attention module, there are 6 fully-connected layers with different hidden sizes: 512, 256, 128, 64, 32 and 5 (output layer).
  + These layers introduce complexity into the model and help transform the information captured in the Bi-LSTMs into final predictions.
* Regularization: The last two dense layers (those with 64 and 32 neurons) are subject to L2 regularization with a regularization parameter of 0.005. This helps prevent overfitting and improve learning curves.

Without regularization:

Where:

* + n is the number of training examples.
  + y\_i is the actual (real) value for observation i.
  + ŷ\_i is the model prediction for observation i.

The MSE with L2 Regularization becomes:

Where:

* + λ is the specified regularization parameter L2 (in this case: 0.005).
  + Σ is the sum on all training examples.
  + j is the index of weights (parameters) in the regularized layer.
  + w\_j represents the j-th weight (parameter) of the layer.
* Loss function: The loss function used to train the model is the Mean Squared Error (MSE). This function evaluates the difference between model predictions and actual values.
* Optimizer: The optimizer used is the Adam with a learning rate of 0.000015. It is responsible for updating the model weights so as to minimize the MSE Loss, allowing the model to learn how to make accurate predictions based on training data.
* Learning Rate Reduction Callback: A callback is used to monitor the loss on the validation set during training and reduce the learning rate by 30% if the loss does not improve for a number of eras (patience=10).

This callback is particularly useful to prevent the model from getting stuck in a plateau during training and to help it converge to a better solution.

The figure shows the model in its entirety:

Image containing text, black and white, receipt, Font

Auto-generated description

**CLASSIC TRANSFORMER**

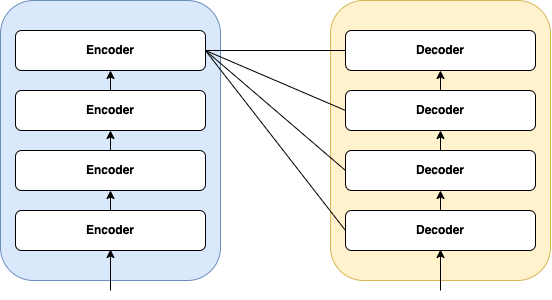
Transformers are another Neural Network architecture and are able to capture long-range content, learning dependencies in Data between distant positions in the Input and Output sequences.

In classical architecture we have 2 fundamental components:

* Encoding component, or a stack of Encoders, all consisting of: a Self Attention Layer and a Feed Forward Neural Network and do not share weights.
* Decoding component, i.e. a stack of Decoders, which use the encoded information of the input sequence to generate the desired output, making use of attention both between the words of the partial output and between the output of the Encoder and the partial output.

Image containing text, diagram, screenshot, Plan

Auto-generated description



**MULTI-HEADED ATTENTION**

"Multi-Head Self-Attention" is a key component of the Transformer architecture that allows the model to capture long-range relationships between words in a sequence. This attention technique is "multi-headed" because it is applied in parallel by several heads within the same layer of the Encoder or Decoder. Each head learns different representations of attention from the input sequence, and their outputs are concatenated and combined to produce a final representation.

Here are the steps of Multi-Head Self-Attention:

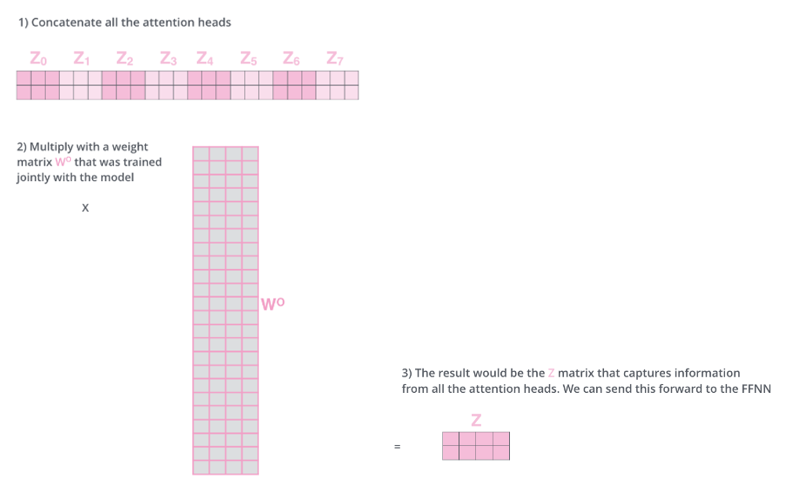
* Query, Key, and Value Preparation: For each head, a set of matricies is created: queries, keys, and values through linear projections of input data. These projections are learned during model training.
  + Queries represent the words of interest for attention,
  + Keys are the words with which to compare queries
  + The values contain the information associated with the keys.
* Calculation of Activation Scores: For each head, a score is associated with each word of the sentence in the INPUT with respect to the word being examined.

Activation scores are calculated using the scalar product between the queries of the current tokens and all the keys of the other tokens.

* Application of Softmax: Activation scores are passed through a softmax function, which normalizes them.
  + This assigns relative weight to each word in the sequence based on its relevance to queries.
* Calculated Weighted Attention: Normalized activation scores are used to weigh values.
  + In this step, you combine information from different words in the sequence based on their relevance to queries.
* Concatenation of Head Attentions: The Z0, Z1, ..., Z7 outputs of all heads are concatenated to form a multi-head representation of attention. This means that the model can capture different relationships and patterns in the sequence.
* Linear Projection: The concatenated representation of attentions is again projected into a lower dimensional space through a further linear projection, i.e. the combination of Zn is multiplied by the weight matrix W0. This projection helps the model combine information from all heads consistently.
* Final Output: The final Z output of the multi-head attention is now ready to be used in further steps of the Transformer architecture, such as feedforward networking or residual connection with layer normalization.

Image that contains diagram

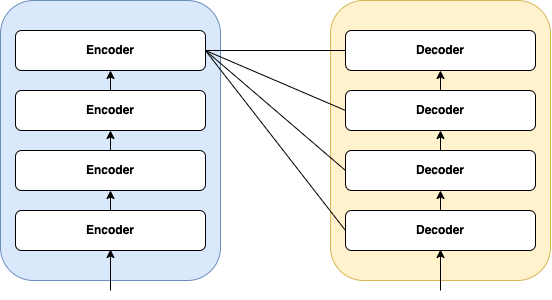
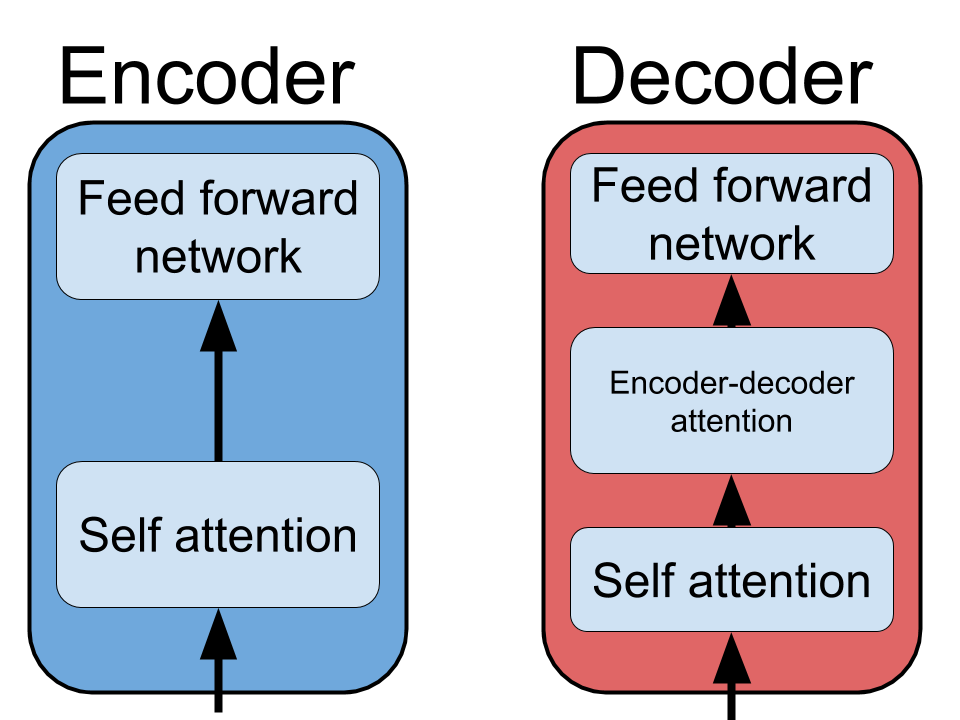
Auto-generated description



**SIMPLIFIED TRANSFORMER MODEL**

The second model analysed in this report is a simplified Transformer with only the Encoder stack part and no Decoder stack part.

This means that the model is primarily designed to encode and process input, but is not designed to generate sequential or textual output based on this encoding. This decision was made because the main objective is the prediction of time series based on numerical data, where there is no need to generate sequences of text or other sequential output. The encoder stack can be used to extract meaningful patterns and features from historical data, which can then be used to make predictions.

Here is a detailed description of the Transformer model with 2 encoders:

* Data Dimensions: num\_features: The number of features in the dataset is set to 5, which means each data example has 5 features or columns.
* Creation of the Sequential Model: A sequential model transformer\_model is created, which is a sequence of data processing layers.
* Adding Layers to the Model: input\_layer: This layer defines the input to the model with a variable shape (None, num\_features) to adapt to the entire dataset. The input represents the model's input data.
* First Encoder: This is the first encoder of the model. It uses a Self-Attention layer with the specified number of heads (num\_heads) and head dimension (head\_dim). Subsequently, it applies two dense layers with ReLU activation. This encoder captures long-term relationships between input features.
* Second Encoder: This is the second encoder of the model. It also utilizes a Self-Attention layer followed by two dense layers with ReLU activation. This encoder can further capture and refine relationships between features.
* Concatenating the Output of Two Encoders to combine the information captured by both. This concatenation step helps create an overall representation of the input data.
* Output Layer: This final layer generates the model's output with a number of features equal to the input (num\_features).
* Model Compilation: The model is compiled using a mean squared error loss function ('mse') and the Adam optimizer with a specified learning rate.
* Adding a Callback to reduce the learning rate during training if the val\_loss stops improving. This helps stabilize training.
* Training the Model: The model is trained using the training data (X\_train\_3d, y\_train\_3d) for a total of 150 epochs with a batch size of 64. During training, val\_loss is monitored, and the callback may reduce the learning rate if necessary.

In summary, the Transformer model with 2 encoders uses multi-head attention and dense layers to process input data and generate predicted output. This model can be used for time series prediction tasks, such as predicting stock prices from Yahoo Finance database, where the goal is to forecast continuous values based on a set of input features. Its architecture allows it to capture complex relationships between data features.

The figure shows the model in its entirety:

Immagine che contiene testo, diagramma, Carattere, ricevuta

Descrizione generata automaticamente

**MODELLO HYBRID – MIXTURE MODEL**

The hybrid mixture model is a machine learning model that combines two popular deep learning models for time series prediction: bi-LSTM models and transformer models and works as follows:

* Time series data is first fed to the bi-LSTM model. The bi-LSTM model learns time dependencies in the data and produces a forecast for the future value of the time series.
* The time series data is then fed to the transformer model. The transformer model learns long-range relationships in the data and produces a forecast for the future value of the time series.
* The predictions of the bi-LSTM model and the transformer model are then combined using a weighted average via the Average() function. The result is a union of the forecasting capabilities of both approaches. The hybrid model is then trained to minimize the mean square error between its predictions and real data. During training, a strategy of dynamic reduction of the learning rate is used to ensure stable convergence.
* Once trained, the hybrid model is evaluated using several evaluation metrics, including RMSE, R-squared, and MAE. These metrics provide a clear indication of its performance in stock price forecasting.
* Finally, the results are displayed through graphs that show the comparison between the forecasts of the hybrid model and the real data for each of the five financial features. This allows a visual analysis of the performance of the model.

The mixture model is a very powerful model for time series prediction and in fact is able to learn both time dependencies and long-range relationships in time series data, which makes it more accurate than many other time series prediction models. Let's see a graph that shows its architecture:

**Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente**

**HYBRID MODEL – WEIGHTED ATTENTION**

The hybrid model with weighted attention represents an advanced implementation to address the challenge of financial time series forecasting, combining Bi-LSTM and simplified Transformer approaches with Multi-headed attention. This hybrid model aims to harness the forces of both models to achieve more accurate and robust predictions.

First you define the inputs for the hybrid model, which accept data in sequential form, with each sequence containing 5 financial features: Close, Open, High, Low and Volume. This data is then passed to two distinct branches of the model.

* The first branch involves the Bi-LSTM model with Multi-attention, which was previously described. This model uses multi-head attention architecture to capture complex relationships in sequential data, focusing on different parts of sequences in parallel.
* The second branch involves the simplified Transformer with 2 Encoder, which also uses the multi-head attention architecture previously described.

After obtaining the exits from the two branches, we move on to calculate the weights of attention. This step includes a dense neural network with a single output neuron, and produces weights that indicate how reliable the outputs of the two branches with sigmoid activation are. In other words, determine how much each branch contributes to the final forecasts.

In a consecutive manner, the outputs of the Bi-LSTM and Transformer models are weighed using attention weights, then the first branch is multiplied by the attention weights obtained, while the second branch is multiplied by the opposite of the attention weights. All this allows the model to assign more importance to one of the two branches based on its relevance and accuracy for a given timeline.

Subsequently, the weighted outputs of the two branches are combined together through the sum operation.

To further enhance the model's ability to capture complex relationships in data, several dense layers are added with various units but all with the same ReLU activation function. These additional layers help the model to perform further processing and adaptations of the combined outputs.

Finally, the hybrid model is compiled using a loss function based on the mean square error (MSE), optimized with the Adam optimizer and, like the other models described above, a dynamic learning rate reduction strategy is used during training to ensure stable convergence.

It can be said that this hybrid model is able to learn both time dependencies and long-range relationships in time series data, which makes it more accurate than many other time series prediction models. Its ability to combine forecasts from two different approaches makes it a powerful tool for obtaining accurate forecasts in financial markets.

This can be seen best in the following figure:

Immagine che contiene testo, diagramma, Piano, Disegno tecnico

Descrizione generata automaticamente

**MODEL DIAGNOSTICS - EVALUATION METRICS**

In this project the following metrics are used for the evaluation of both the models and the results produced. Since this is a regression task, the metrics are:

* **MSE Loss**: This metric represents the value of the loss function calculated on the test set.
  + Loss is a measure of the discrepancy between model predictions and actual values, a lower loss value indicates better adaptability of the model to test data.
  + It is calculated as part of the training process and evaluated on the test data.
* **Mean Absolute Error (MAE)** measures the absolute mean error between model predictions and actual values in test data. It is an error metric that indicates the average deviation of forecasts from the real value. It is calculated as the average of the absolute differences between the predictions and the actual values for each sample in the test data.
* **Root Mean Squared Error (RMSE):**  The RMSE is a measure of error that takes into account both the difference between the predictions of the model and the actual values and their dispersion. It is the square root of the square mean of the differences between the predictions and the actual values in the test data.
  + A lower RMSE indicates better forecast accuracy.
* **R-squared (R²):** Also known as the coefficient of determination, it measures how well the model fits the data compared to an average-based model. R² ranges from 0 to 1, where:
  + 1 indicates a perfect model that fits perfectly with the data.
  + 0 indicates that the model offers no benefit over a simple average-based forecast.
  + It is calculated by comparing the variation explained by the model with the total change in the data.
* **Learning Curves** are graphs that show how training and validation metrics vary as the number of training eras increase. They can be useful for assessing whether the model is learning from the data and whether there is overfitting or underfitting. Typically, a good balance is sought between training loss and validation and MAE.
* **Graphs & Tables:** the last metrics are the graphs and tables that make us realise how well each model, for each dataset, works on the validation and test set predictions.

**RESULTS - Model BILSTM**

**APPLE DATASET**

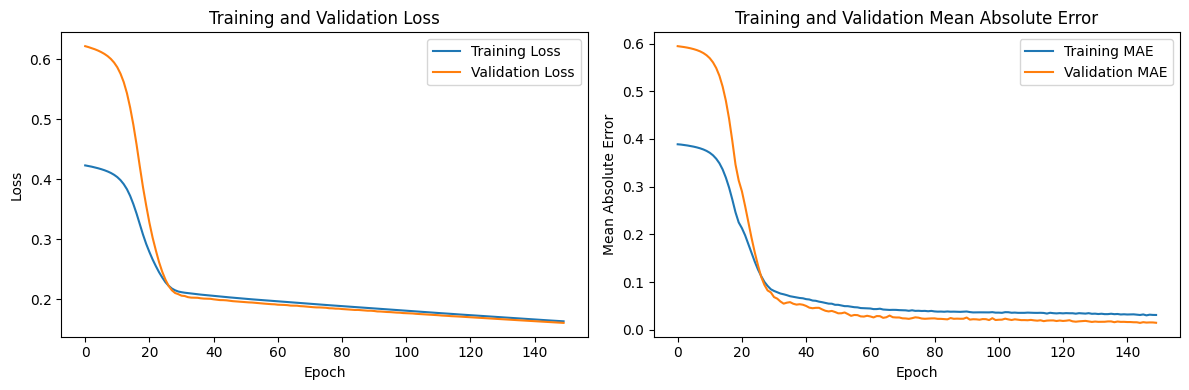
The training is done with 150 eras and a batch-size of 64, learning\_rate=0.000015 with a 30% reduction callback.

* **Test Loss: 0.09** The value of the Test Loss is 0.09, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.0138** which indicates that, on average, the predictions of the model differ only by 0.0138 from the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.0198** The RMSE takes into account both the difference between the model predictions and the actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.9091** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.9091 suggests that the model explains about 90.91% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.0912 at epoch 150**
  + **Best Training Loss: 0.0914 at epoch 150**

During the training process, we achieved the lowest validation loss, which is our primary metric for evaluating model performance, at the 150th epoch, with a loss value of 0.0912. This means that the model has achieved its best performance in generalizing to new data at this stage.

However, it is interesting to note that the slightest loss of training was also obtained at the time 150, but with a slightly higher value of 0.0912. This may indicate a slight overlap between the model and the training dataset, but the difference between training and validation losses is minimal.

Both results suggest that our model performed well in learning patterns in data and in generalizing to new examples.



Let's see a graphical representation of the results of the predictions:

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.713835 0.743071 0.734673 0.718817 0.165831**

**1 0.685644 0.691561 0.699523 0.688478 0.195604**

**2 0.716304 0.687773 0.709401 0.690409 0.155691**

**3 0.747171 0.730124 0.739578 0.718541 0.179323**

**4 0.754236 0.740867 0.746390 0.740399 0.111776**

**.. ... ... ... ... ...**

**155 0.662666 0.632476 0.656812 0.637110 0.139451**

**156 0.677618 0.662225 0.673025 0.660208 0.112703**

**157 0.697442 0.682195 0.688420 0.683376 0.109540**

**158 0.676933 0.688187 0.685014 0.674274 0.143617**

**159 0.645723 0.668285 0.664986 0.647314 0.196739**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.728675 0.721067 0.729000 0.726633**

**1 0.693365 0.687022 0.694328 0.690700**

**2 0.702554 0.695340 0.703204 0.700272**

**3 0.735461 0.727859 0.735731 0.733416**

**4 0.745444 0.736892 0.745442 0.743848**

**.. ... ... ... ...**

**155 0.649117 0.643022 0.650531 0.646290**

**156 0.669406 0.662387 0.670386 0.667001**

**157 0.688757 0.681259 0.689448 0.686602**

**158 0.682431 0.675562 0.683342 0.679981**

**159 0.658653 0.653202 0.660207 0.655621**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.720763 0.669180 0.722548 0.672757 0.337435**

**1 0.704301 0.703336 0.700341 0.701027 0.168294**

**2 0.685849 0.716558 0.708583 0.681790 0.123835**

**3 0.647301 0.674345 0.686240 0.653313 0.157312**

**4 0.605117 0.626898 0.622411 0.610219 0.168232**

**.. ... ... ... ...**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.699876 0.695549 0.701310 0.696128**

**1 0.703985 0.696927 0.704663 0.701637**

**2 0.698610 0.691114 0.699218 0.696478**

**3 0.666671 0.660446 0.667896 0.663933**

**4 0.618296 0.613411 0.620205 0.614662**

**.. ... ... ... ...**

**AMAZON DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.000015 with a 30% reduction callback.

* **Test Loss: 0.16** The value of the Test Loss is 0.16, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.013**  which indicates that, on average, the predictions of the model differ only by 0.013 from the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.0197** The RMSE takes into account both the difference between the model predictions and the actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.91** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.91 suggests that the model explains about 91% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.1360 at epoch 150**
  + **Best Training Loss: 0.1356 at epoch 150**

Immagine che contiene testo, linea, diagramma, Diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, linea, Diagramma, schermata

Descrizione generata automaticamente

Immagine che contiene testo, linea, Diagramma, schermata

Descrizione generata automaticamente

Immagine che contiene testo, linea, Diagramma, Carattere

Descrizione generata automaticamente

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.608155 0.642086 0.629386 0.618783 0.174568**

**1 0.573105 0.607903 0.608230 0.575465 0.192224**

**2 0.625739 0.578119 0.613317 0.586244 0.197324**

**3 0.680512 0.631865 0.662426 0.637949 0.229239**

**4 0.719987 0.671252 0.703772 0.684179 0.188302**

**.. ... ... ... ... ...**

**155 0.357994 0.311198 0.343336 0.320641 0.129492**

**156 0.362767 0.360568 0.350942 0.340633 0.108626**

**157 0.370213 0.357429 0.359785 0.363620 0.113403**

**158 0.323053 0.322708 0.341054 0.323152 0.174235**

**159 0.278185 0.302922 0.291327 0.274957 0.381245**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.625281 0.618416 0.628225 0.623481**

**1 0.592706 0.585389 0.596098 0.589135**

**2 0.602143 0.594907 0.605444 0.598525**

**3 0.654909 0.648307 0.657853 0.648794**

**4 0.695597 0.689617 0.697944 0.693272**

**.. ... ... ... ...**

**155 0.338675 0.328333 0.342108 0.338664**

**156 0.358222 0.349026 0.362012 0.359996**

**157 0.366337 0.357188 0.370166 0.367924**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.206109 0.150626 0.194725 0.157995 0.699671**

**1 0.196849 0.204372 0.203377 0.187743 0.277902**

**2 0.142911 0.208462 0.200620 0.142542 0.462154**

**3 0.098329 0.145014 0.135585 0.103426 0.402206**

**4 0.071408 0.098878 0.095271 0.074548 0.405343**

**.. ... ... ... ... ...**

**196 0.540236 0.526276 0.531025 0.537952 0.085817**

**197 0.560857 0.534837 0.543005 0.545196 0.100513**

**198 0.533172 0.551484 0.549565 0.540174 0.085644**

**199 0.508351 0.524278 0.511438 0.518732 0.081880**

**200 0.497945 0.507822 0.500219 0.504438 0.104617**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.241667 0.185762 0.204271 0.244494**

**1 0.194225 0.168183 0.183012 0.205084**

**2 0.199075 0.158829 0.173950 0.205059**

**3 0.154145 0.111123 0.122777 0.146449**

**4 0.132190 0.086538 0.097194 0.111161**

**.. ... ... ... ...**

**196 0.524018 0.517447 0.528206 0.527292**

**197 0.536978 0.529727 0.540788 0.539982**

**198 0.533473 0.527549 0.537636 0.536634**

**199 0.505819 0.498413 0.510145 0.509362**

**200 0.494276 0.483518 0.498075 0.497800**

**GOOGLE DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.000015 with a 30% reduction callback.

* **MSE Test Loss: 0.129** The value of the Test Loss is 0.129, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.008** which indicates that, on average, the predictions of the model differ only by 0.008 from the real observations, being a very low error measure and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.0113** The RMSE takes into account both the difference between model predictions and actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.993** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.993 suggests that the model explains about 99.3% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.129 at epoch 150**
  + **Best Training Loss: 0.13 at epoch 150**

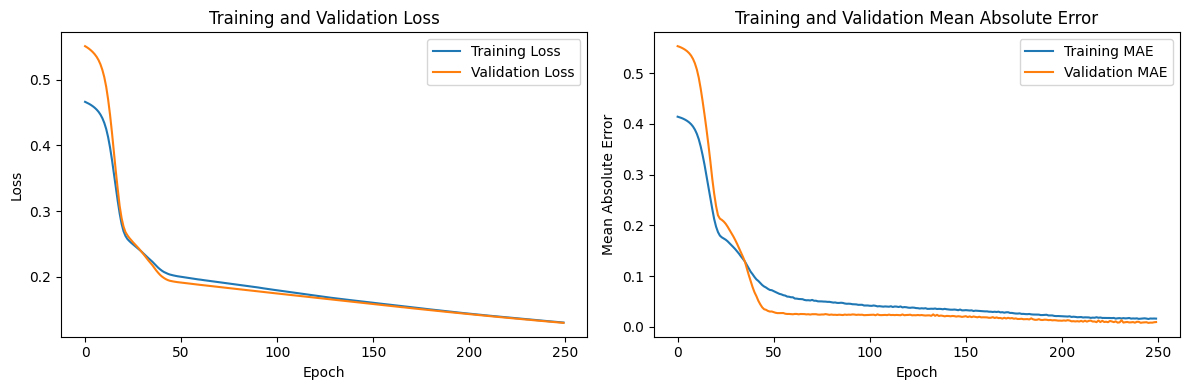


Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

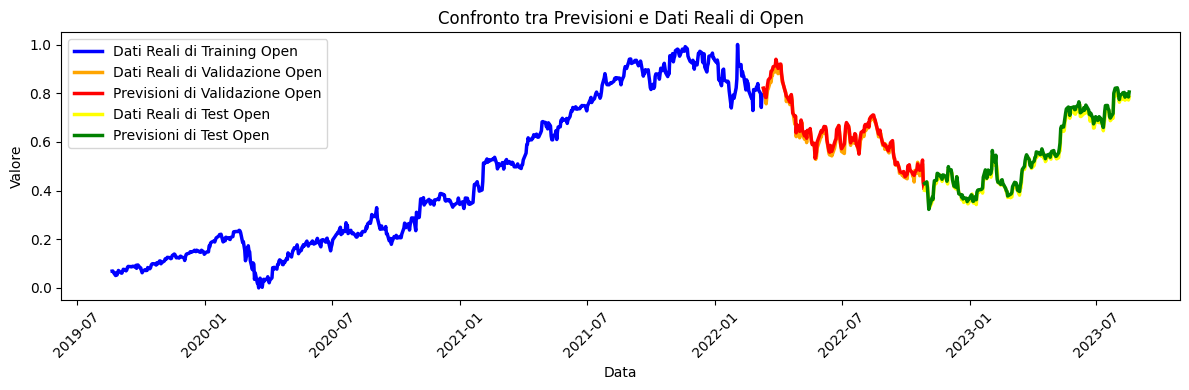
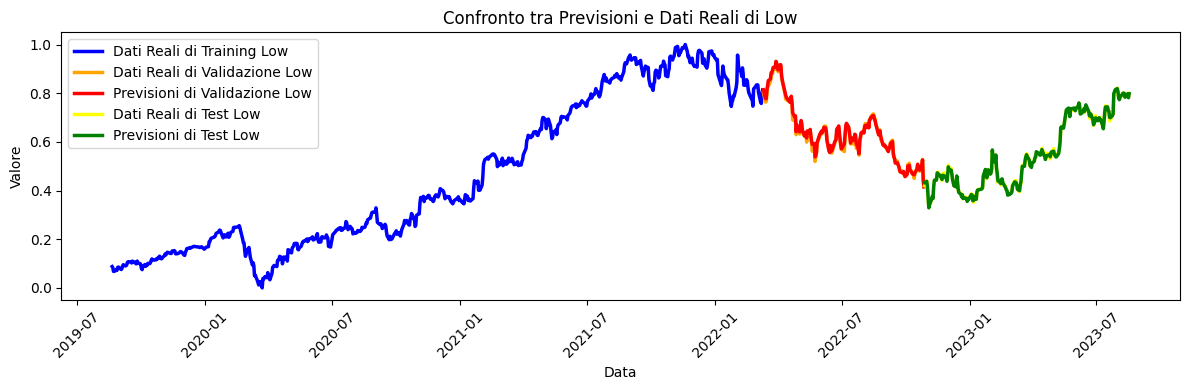


Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente



Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.793278 0.819625 0.818337 0.802531 0.216415**

**1 0.755124 0.785027 0.786125 0.763387 0.256629**

**2 0.784952 0.756437 0.780961 0.764950 0.257047**

**3 0.826125 0.789338 0.813983 0.791489 0.276483**

**4 0.835423 0.812774 0.823812 0.826949 0.187734**

**.. ... ... ... ... ...**

**155 0.497037 0.460778 0.487690 0.479315 0.242704**

**156 0.512260 0.497430 0.502710 0.500180 0.195293**

**157 0.532285 0.509648 0.523007 0.527596 0.252846**

**158 0.428993 0.443613 0.456431 0.442424 0.710615**

**159 0.406312 0.418875 0.422230 0.415512 0.518371**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.811963 0.820638 0.810010 0.814165**

**1 0.774065 0.783113 0.775160 0.778245**

**2 0.773107 0.782076 0.774059 0.777190**

**3 0.807808 0.817387 0.809750 0.812242**

**4 0.828950 0.837004 0.825038 0.829546**

**.. ... ... ... ...**

**155 0.476523 0.482924 0.479499 0.483903**

**156 0.499046 0.504844 0.498697 0.504321**

**157 0.518423 0.525326 0.521619 0.525651**

**158 0.436830 0.432808 0.441044 0.444344**

**159 0.404801 0.409679 0.416176 0.415696**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.446975 0.400902 0.439381 0.419775 0.316532**

**1 0.427359 0.433717 0.434205 0.440509 0.252389**

**2 0.384857 0.431799 0.432328 0.400695 0.399335**

**3 0.349813 0.384544 0.382954 0.366223 0.403000**

**4 0.313237 0.338451 0.334747 0.330339 0.457552**

**.. ... ... ... ... ...**

**196 0.790157 0.771184 0.780177 0.786943 0.090857**

**197 0.807117 0.777727 0.795096 0.795410 0.116552**

**198 0.791179 0.795296 0.795908 0.797718 0.086209**

**199 0.779327 0.771971 0.784825 0.784020 0.116796**

**200 0.793120 0.783785 0.800992 0.798031 0.184120**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.421118 0.429161 0.429979 0.432650**

**1 0.429497 0.435723 0.433729 0.437950**

**2 0.404371 0.412814 0.413808 0.417057**

**3 0.362146 0.369724 0.370647 0.375162**

**4 0.321136 0.322456 0.328475 0.328998**

**.. ... ... ... ...**

**196 0.778879 0.783679 0.777549 0.783336**

**197 0.792883 0.798012 0.789483 0.795110**

**198 0.791809 0.796686 0.791159 0.796957**

**199 0.779143 0.784319 0.775512 0.781133**

**200 0.796947 0.804801 0.793166 0.797920**

**MICROSOFT DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.000015 with a 30% reduction callback.

* **Test Loss: 0.13** The value of the Test Loss is 0.13, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE):** 0.0056 which indicates that, on average, the predictions of the model differ only by 0.0056 from the real observations, being a very low error measure and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.0082** The RMSE takes into account both the difference between model predictions and actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.997** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.997 suggests that the model explains about 99.7% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.1380 at epoch 150**
  + **Best Training Loss: 0.1385 at epoch 150**

Immagine che contiene testo, linea, Diagramma, diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.648740 0.674748 0.665816 0.668016 0.206991**

**1 0.632685 0.641160 0.648041 0.651601 0.246201**

**2 0.680053 0.641204 0.658507 0.664833 0.286922**

**3 0.712074 0.679817 0.687700 0.685158 0.327607**

**4 0.715745 0.698241 0.692198 0.713214 0.247972**

**.. ... ... ... ... ...**

**155 0.480893 0.440164 0.464666 0.463714 0.196658**

**156 0.503583 0.479922 0.485598 0.494634 0.180884**

**157 0.518664 0.495350 0.499438 0.515233 0.292948**

**158 0.433127 0.424428 0.444339 0.443525 0.835618**

**159 0.412915 0.423855 0.424401 0.424063 0.357125**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.664906 0.656778 0.649806 0.672330**

**1 0.643073 0.635150 0.630086 0.650127**

**2 0.660432 0.653107 0.648875 0.668153**

**3 0.690114 0.683516 0.679836 0.698889**

**4 0.704827 0.697561 0.690978 0.713707**

**.. ... ... ... ...**

**155 0.465331 0.454994 0.452606 0.466680**

**156 0.493865 0.483670 0.480123 0.496052**

**157 0.506990 0.498209 0.497533 0.510219**

**158 0.424612 0.425513 0.422480 0.418402**

**159 0.419922 0.411330 0.414210 0.421435**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.453251 0.402698 0.436986 0.425291 0.359660**

**1 0.436709 0.435844 0.429721 0.448481 0.220033**

**2 0.419195 0.439547 0.433267 0.431111 0.245424**

**3 0.383503 0.416891 0.414065 0.397963 0.334204**

**4 0.357629 0.375590 0.366967 0.370407 0.314060**

**.. ... ... ... ... ...**

**196 0.829810 0.817120 0.808105 0.848900 0.174424**

**197 0.843211 0.822101 0.815241 0.852856 0.111867**

**198 0.833569 0.829197 0.819696 0.856584 0.090625**

**199 0.827112 0.819500 0.816798 0.851582 0.133029**

**200 0.811544 0.818354 0.805769 0.835258 0.139372**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.428295 0.419840 0.422661 0.430089**

**1 0.440179 0.429787 0.428605 0.440824**

**2 0.432936 0.422800 0.422449 0.433563**

**3 0.402877 0.393795 0.396419 0.403651**

**4 0.368520 0.358837 0.361569 0.368079**

**.. ... ... ... ...**

**196 0.828189 0.820228 0.808507 0.839466**

**197 0.836146 0.825157 0.810949 0.844178**

**198 0.835988 0.823074 0.807032 0.838445**

**199 0.831597 0.821961 0.808865 0.841778**

**200 0.820692 0.811272 0.798726 0.830987**

**RESULTS - SIMPLIFIED TRANSFORMER MODEL**

**APPLE DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.000015 with a 30% reduction callback.

* **Test Loss: 0.00042** The value of the Test Loss is 0.00042, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.017** which indicates that, on average, the predictions of the model differ only by 0.017 from the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.025** The RMSE takes into account both the difference between model predictions and actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.90** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.90 suggests that the model explains about 97.5% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.0002 at epoch 150**
  + **Best Training Loss: 0.0010 at epoch 150**

**Immagine che contiene testo, linea, Diagramma, diagramma

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.713835 0.743071 0.734673 0.718817 0.165831**

**1 0.685644 0.691561 0.699523 0.688478 0.195604**

**2 0.716304 0.687773 0.709401 0.690409 0.155691**

**3 0.747171 0.730124 0.739578 0.718541 0.179323**

**4 0.754236 0.740867 0.746390 0.740399 0.111776**

**.. ... ... ... ... ...**

**155 0.662666 0.632476 0.656812 0.637110 0.139451**

**156 0.677618 0.662225 0.673025 0.660208 0.112703**

**157 0.697442 0.682195 0.688420 0.683376 0.109540**

**158 0.676933 0.688187 0.685014 0.674274 0.143617**

**159 0.645723 0.668285 0.664986 0.647314 0.196739**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.728931 0.733144 0.734106 0.722639**

**1 0.691755 0.694214 0.695752 0.685189**

**2 0.703901 0.706343 0.706504 0.696370**

**3 0.736845 0.741615 0.742488 0.730297**

**4 0.748822 0.751930 0.751521 0.741521**

**.. ... ... ... ...**

**155 0.649396 0.649018 0.648648 0.641000**

**156 0.670740 0.670496 0.669988 0.662241**

**157 0.690543 0.690696 0.690355 0.682230**

**158 0.682415 0.683642 0.684183 0.674985**

**159 0.655424 0.656325 0.657795 0.648743**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.720763 0.669180 0.722548 0.672757 0.337435**

**1 0.704301 0.703336 0.700341 0.701027 0.168294**

**2 0.685849 0.716558 0.708583 0.681790 0.123835**

**3 0.647301 0.674345 0.686240 0.653313 0.157312**

**4 0.605117 0.626898 0.622411 0.610219 0.168232**

**.. ... ... ... ... ...**

**197 0.883463 0.874187 0.873706 0.876095 0.030926**

**198 0.869676 0.880453 0.872275 0.874302 0.030792**

**199 0.863640 0.868402 0.865872 0.870510 0.039252**

**200 0.846011 0.868471 0.858856 0.849686 0.087595**

**201 0.849372 0.835141 0.842439 0.839206 0.075069**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.694848 0.698681 0.701761 0.689406**

**1 0.704145 0.706498 0.707938 0.697241**

**2 0.698853 0.700843 0.700524 0.691561**

**3 0.664924 0.666357 0.666342 0.657696**

**4 0.615264 0.613983 0.614943 0.607855**

**.. ... ... ... ...**

**197 0.869203 0.884197 0.884305 0.871880**

**198 0.865744 0.880547 0.880743 0.868506**

**199 0.860623 0.874357 0.874373 0.862327**

**200 0.856750 0.869240 0.868934 0.855597**

**201 0.842197 0.852921 0.852034 0.840090**

**AMAZON DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.000015 with a 30% reduction callback.

* **Test Loss: 0.0010** The value of the Test Loss is 0.0010, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.017** which indicates that, on average, the predictions of the model differ only by 0.017 from the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.026** The RMSE takes into account both the difference between model predictions and actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.90** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.90 suggests that the model explains about 93.8% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.0010 at epoch 150**
  + **Best Training Loss: 0.0012 at epoch 150**

**Immagine che contiene Diagramma, linea, testo, diagramma

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, linea, Carattere, Diagramma

Descrizione generata automaticamente**

**Immagine che contiene testo, linea, schermata, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, linea, Carattere, Diagramma

Descrizione generata automaticamente**

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.608155 0.642086 0.629386 0.618783 0.174568**

**1 0.573105 0.607903 0.608230 0.575465 0.192224**

**2 0.625739 0.578119 0.613317 0.586244 0.197324**

**3 0.680512 0.631865 0.662426 0.637949 0.229239**

**4 0.719987 0.671252 0.703772 0.684179 0.188302**

**.. ... ... ... ... ...**

**155 0.357994 0.311198 0.343336 0.320641 0.129492**

**156 0.362767 0.360568 0.350942 0.340633 0.108626**

**157 0.370213 0.357429 0.359785 0.363620 0.113403**

**158 0.323053 0.322708 0.341054 0.323152 0.174235**

**159 0.278185 0.302922 0.291327 0.274957 0.381245**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.618943 0.622437 0.619283 0.609189**

**1 0.588953 0.588247 0.585736 0.573567**

**2 0.595621 0.593102 0.589960 0.579651**

**3 0.649607 0.644868 0.643718 0.633654**

**4 0.687446 0.690440 0.687444 0.681601**

**.. ... ... ... ...**

**155 0.343387 0.331315 0.327748 0.313075**

**156 0.361322 0.354582 0.350532 0.336423**

**157 0.368497 0.361335 0.357599 0.343354**

**158 0.342031 0.324356 0.322419 0.305487**

**159 0.317273 0.270743 0.273154 0.257589**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.206109 0.150626 0.194725 0.157995 0.699671**

**1 0.196849 0.204372 0.203377 0.187743 0.277902**

**2 0.142911 0.208462 0.200620 0.142542 0.462154**

**3 0.098329 0.145014 0.135585 0.103426 0.402206**

**4 0.071408 0.098878 0.095271 0.074548 0.405343**

**.. ... ... ... ... ...**

**196 0.540236 0.526276 0.531025 0.537952 0.085817**

**197 0.560857 0.534837 0.543005 0.545196 0.100513**

**198 0.533172 0.551484 0.549565 0.540174 0.085644**

**199 0.508351 0.524278 0.511438 0.518732 0.081880**

**200 0.497945 0.507822 0.500219 0.504438 0.104617**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.236522 0.143898 0.151655 0.148545**

**1 0.232979 0.191248 0.193174 0.180704**

**2 0.223054 0.159370 0.162425 0.157363**

**3 0.173369 0.112160 0.115817 0.111746**

**4 0.143413 0.079646 0.084167 0.080329**

**.. ... ... ... ...**

**196 0.522161 0.532163 0.524408 0.516766**

**197 0.534618 0.543129 0.535777 0.528139**

**198 0.532996 0.544696 0.536666 0.528930**

**199 0.505895 0.515704 0.508355 0.499709**

**200 0.495634 0.500888 0.495065 0.484844**

**MICROSOFT DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.000015 with a 30% reduction callback.

* **Test Loss: 0.00036** The value of the Test Loss is 0.00036, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE):** 0.0064 which indicates that, on average, the model's predictions differ by only 0.0064 of the actual observations, being a very low error measure and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.009** The RMSE takes into account both the difference between the model predictions and the actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.95** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.95 suggests that the model explains about 99.6% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.0003 at epoch 150**
  + **Best Training Loss: 0.0008 at epoch 150**

Immagine che contiene Diagramma, linea, testo, diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.648740 0.674748 0.665816 0.668016 0.206991**

**1 0.632685 0.641160 0.648041 0.651601 0.246201**

**2 0.680053 0.641204 0.658507 0.664833 0.286922**

**3 0.712074 0.679817 0.687700 0.685158 0.327607**

**4 0.715745 0.698241 0.692198 0.713214 0.247972**

**.. ... ... ... ... ...**

**155 0.480893 0.440164 0.464666 0.463714 0.196658**

**156 0.503583 0.479922 0.485598 0.494634 0.180884**

**157 0.518664 0.495350 0.499438 0.515233 0.292948**

**158 0.433127 0.424428 0.444339 0.443525 0.835618**

**159 0.412915 0.423855 0.424401 0.424063 0.357125**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.665786 0.662394 0.646771 0.672456**

**1 0.642519 0.637775 0.627744 0.649906**

**2 0.661176 0.653576 0.647690 0.667526**

**3 0.695225 0.684914 0.681184 0.699518**

**4 0.709127 0.701731 0.689821 0.713884**

**.. ... ... ... ...**

**155 0.457123 0.455340 0.448387 0.468911**

**156 0.486771 0.486388 0.475998 0.498311**

**157 0.500989 0.494232 0.495756 0.510667**

**158 0.404895 0.366834 0.452947 0.420442**

**159 0.412581 0.402281 0.417539 0.423524**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.453251 0.402698 0.436986 0.425291 0.359660**

**1 0.436709 0.435844 0.429721 0.448481 0.220033**

**2 0.419195 0.439547 0.433267 0.431111 0.245424**

**3 0.383503 0.416891 0.414065 0.397963 0.334204**

**4 0.357629 0.375590 0.366967 0.370407 0.314060**

**.. ... ... ... ... ...**

**196 0.829810 0.817120 0.808105 0.848900 0.174424**

**197 0.843211 0.822101 0.815241 0.852856 0.111867**

**198 0.833569 0.829197 0.819696 0.856584 0.090625**

**199 0.827112 0.819500 0.816798 0.851582 0.133029**

**200 0.811544 0.818354 0.805769 0.835258 0.139372**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.419052 0.408193 0.424193 0.430493**

**1 0.433346 0.430118 0.426659 0.444524**

**2 0.426911 0.422608 0.421962 0.437626**

**3 0.396202 0.387348 0.400203 0.407403**

**4 0.361626 0.352773 0.365867 0.373046**

**.. ... ... ... ...**

**196 0.838359 0.829308 0.808218 0.839645**

**197 0.839057 0.835688 0.812166 0.847899**

**198 0.836663 0.835793 0.812221 0.848216**

**199 0.837053 0.831463 0.808332 0.842428**

**200 0.826779 0.821444 0.798044 0.831632**

**GOOGLE DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.000015 with a 30% reduction callback.

* **Test Loss: 0.00065** The value of the Test Loss is 0.00065, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.017** which indicates that, on average, the predictions of the model differ only by 0.017 from the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.025** The RMSE takes into account both the difference between model predictions and actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.959** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.959 suggests that the model explains about 97.5% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.0006 at epoch 150**
  + **Best Training Loss: 0.0009 at epoch 150**

**Immagine che contiene Diagramma, linea, testo, diagramma

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.793278 0.819625 0.818337 0.802531 0.216415**

**1 0.755124 0.785027 0.786125 0.763387 0.256629**

**2 0.784952 0.756437 0.780961 0.764950 0.257047**

**3 0.826125 0.789338 0.813983 0.791489 0.276483**

**4 0.835423 0.812774 0.823812 0.826949 0.187734**

**.. ... ... ... ... ...**

**155 0.497037 0.460778 0.487690 0.479315 0.242704**

**156 0.512260 0.497430 0.502710 0.500180 0.195293**

**157 0.532285 0.509648 0.523007 0.527596 0.252846**

**158 0.428993 0.443613 0.456431 0.442424 0.710615**

**159 0.406312 0.418875 0.422230 0.415512 0.518371**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.804587 0.804279 0.804469 0.811435**

**1 0.763823 0.769363 0.767974 0.776097**

**2 0.761102 0.766948 0.765184 0.773202**

**3 0.799354 0.801515 0.802260 0.807258**

**4 0.821779 0.817441 0.818665 0.825357**

**.. ... ... ... ...**

**155 0.461783 0.468538 0.474460 0.474518**

**156 0.483237 0.491109 0.494012 0.498542**

**157 0.503421 0.510145 0.515701 0.516631**

**158 0.432831 0.411814 0.451818 0.424733**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.446975 0.400902 0.439381 0.419775 0.316532**

**1 0.427359 0.433717 0.434205 0.440509 0.252389**

**2 0.384857 0.431799 0.432328 0.400695 0.399335**

**3 0.349813 0.384544 0.382954 0.366223 0.403000**

**4 0.313237 0.338451 0.334747 0.330339 0.457552**

**.. ... ... ... ... ...**

**196 0.790157 0.771184 0.780177 0.786943 0.090857**

**197 0.807117 0.777727 0.795096 0.795410 0.116552**

**198 0.791179 0.795296 0.795908 0.797718 0.086209**

**199 0.779327 0.771971 0.784825 0.784020 0.116796**

**200 0.793120 0.783785 0.800992 0.798031 0.184120**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.409871 0.413321 0.424450 0.418402**

**1 0.418054 0.423210 0.431044 0.429961**

**2 0.400728 0.397705 0.416150 0.405134**

**3 0.361244 0.356001 0.376583 0.364764**

**4 0.322802 0.311892 0.338085 0.323323**

**.. ... ... ... ...**

**196 0.771890 0.769505 0.767231 0.779359**

**197 0.785209 0.782458 0.781168 0.791500**

**198 0.787247 0.782908 0.781632 0.793137**

**199 0.770318 0.769418 0.767249 0.778290**

**200 0.786584 0.786802 0.785654 0.794153**

**RESULTS – HYBRID MIXTURE MODEL**

This hybrid model combines the predictions of two previously evaluated models and are known for their good results, so there is no need to perform a further evaluation of the hybrid model.

When combining predictions from different models, it is important to ensure that the base models have good individual performance, as the quality of the predictions of the base models will affect the performance of the hybrid model.

**APPLE DATASET** With 150 eras and a batch-size of 64, learning\_rate=0.00001

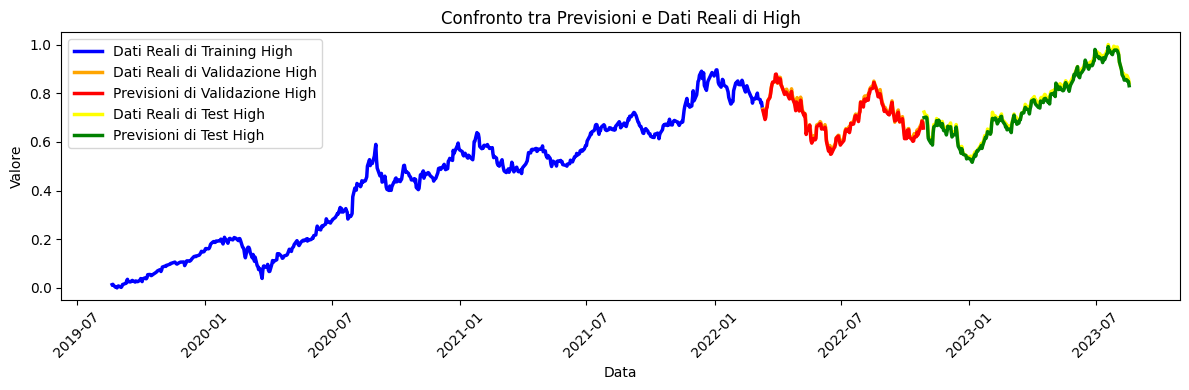
* **Test Mean Absolute Error (MAE):** 0.00987 which indicates that, on average, the model's predictions differ by only 0.00987 of the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.0144** The RMSE takes into account both the difference between the model predictions and the actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.911** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.911 suggests that the model explains about 95.1% of the variation in the data, which is very good for its prediction capabilities.

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente**

****

**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.713835 0.743071 0.734673 0.718817 0.165831**

**1 0.685644 0.691561 0.699523 0.688478 0.195604**

**2 0.716304 0.687773 0.709401 0.690409 0.155691**

**3 0.747171 0.730124 0.739578 0.718541 0.179323**

**4 0.754236 0.740867 0.746390 0.740399 0.111776**

**.. ... ... ... ... ...**

**155 0.662666 0.632476 0.656812 0.637110 0.139451**

**156 0.677618 0.662225 0.673025 0.660208 0.112703**

**157 0.697442 0.682195 0.688420 0.683376 0.109540**

**158 0.676933 0.688187 0.685014 0.674274 0.143617**

**159 0.645723 0.668285 0.664986 0.647314 0.196739**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.733621 0.733133 0.728611 0.731569**

**1 0.694003 0.694504 0.691707 0.690420**

**2 0.703942 0.703938 0.700130 0.701844**

**3 0.738519 0.738137 0.734621 0.736679**

**4 0.752923 0.749097 0.742604 0.749750**

**.. ... ... ... ...**

**155 0.648231 0.648743 0.645185 0.645128**

**156 0.671043 0.670467 0.665690 0.668392**

**157 0.691796 0.690258 0.685096 0.688876**

**158 0.684446 0.684434 0.679999 0.681696**

**159 0.658016 0.658419 0.656598 0.652942**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.720763 0.669180 0.722548 0.672757 0.337435**

**1 0.704301 0.703336 0.700341 0.701027 0.168294**

**2 0.685849 0.716558 0.708583 0.681790 0.123835**

**3 0.647301 0.674345 0.686240 0.653313 0.157312**

**4 0.605117 0.626898 0.622411 0.610219 0.168232**

**.. ... ... ... ... ...**

**197 0.883463 0.874187 0.873706 0.876095 0.030926**

**198 0.869676 0.880453 0.872275 0.874302 0.030792**

**199 0.863640 0.868402 0.865872 0.870510 0.039252**

**200 0.846011 0.868471 0.858856 0.849686 0.087595**

**201 0.849372 0.835141 0.842439 0.839206 0.075069**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.693421 0.694137 0.699402 0.686060**

**1 0.705701 0.705862 0.701802 0.703038**

**2 0.703772 0.702178 0.697177 0.700938**

**3 0.668544 0.668919 0.664976 0.665144**

**4 0.616910 0.617613 0.615149 0.611531**

**.. ... ... ... ...**

**197 0.876768 0.868114 0.855779 0.875297**

**198 0.874608 0.865987 0.853153 0.872718**

**199 0.868660 0.860329 0.848122 0.866714**

**200 0.866700 0.858267 0.848103 0.863497**

**201 0.848276 0.840677 0.830758 0.845644**

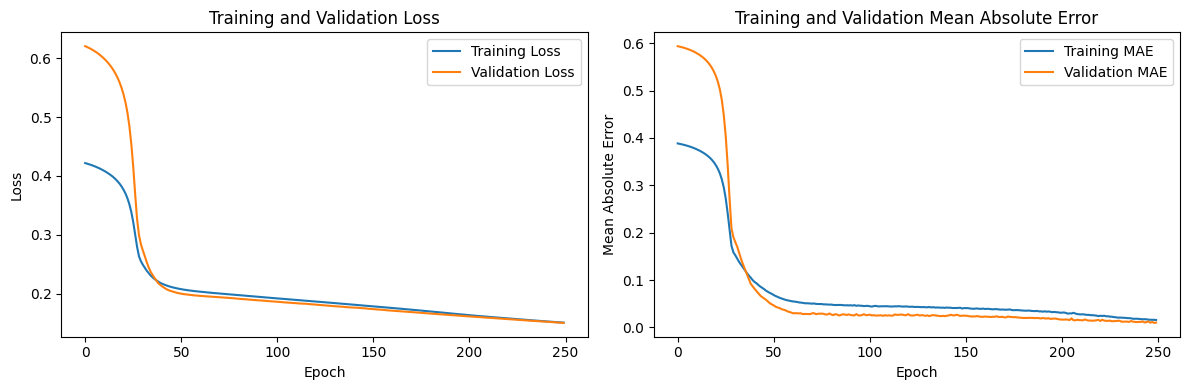
**Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente**

**RESULTS - HYBRID model with WEIGHTED ATTENTION**

**APPLE DATASET** The training is done with 150 eras and a batch-size of 64, learning\_rate=0.00001 with a 30% reduction callback.

* **Test Loss: 0.10** The value of the Test Loss is 0.10, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.012** which indicates that, on average, the predictions of the model differ only by 0.012 from the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.016** The RMSE takes into account both the difference between the model predictions and the actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.90** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.95 suggests that the model explains about 95% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.1507 at epoch 150**
  + **Best Training Loss: 0.1511 at epoch 150**

****

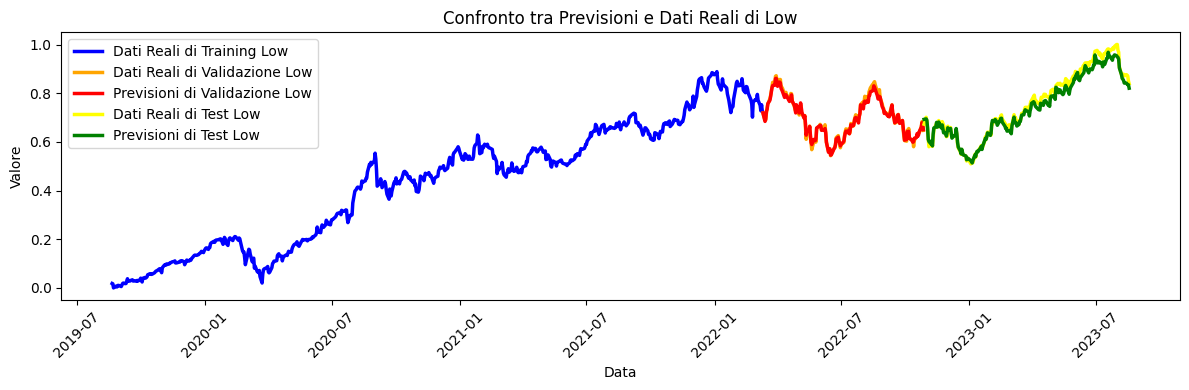
**Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente**

**Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente**

****

****

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.713835 0.743071 0.734673 0.718817 0.165831**

**1 0.685644 0.691561 0.699523 0.688478 0.195604**

**2 0.716304 0.687773 0.709401 0.690409 0.155691**

**3 0.747171 0.730124 0.739578 0.718541 0.179323**

**4 0.754236 0.740867 0.746390 0.740399 0.111776**

**.. ... ... ... ... ...**

**155 0.662666 0.632476 0.656812 0.637110 0.139451**

**156 0.677618 0.662225 0.673025 0.660208 0.112703**

**157 0.697442 0.682195 0.688420 0.683376 0.109540**

**158 0.676933 0.688187 0.685014 0.674274 0.143617**

**159 0.645723 0.668285 0.664986 0.647314 0.196739**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.723958 0.725188 0.725915 0.718713**

**1 0.690616 0.694547 0.690588 0.684018**

**2 0.699495 0.700241 0.700737 0.693655**

**3 0.731317 0.734464 0.732854 0.725743**

**4 0.736109 0.733689 0.741889 0.733699**

**.. ... ... ... ...**

**155 0.648578 0.648231 0.647624 0.641169**

**156 0.665577 0.663296 0.667130 0.660224**

**157 0.683038 0.680600 0.685904 0.678595**

**158 0.679690 0.679184 0.680154 0.673383**

**159 0.657525 0.660891 0.656741 0.650352**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.720763 0.669180 0.722548 0.672757 0.337435**

**1 0.704301 0.703336 0.700341 0.701027 0.168294**

**2 0.685849 0.716558 0.708583 0.681790 0.123835**

**3 0.647301 0.674345 0.686240 0.653313 0.157312**

**4 0.605117 0.626898 0.622411 0.610219 0.168232**

**.. ... ... ... ... ...**

**197 0.883463 0.874187 0.873706 0.876095 0.030926**

**198 0.869676 0.880453 0.872275 0.874302 0.030792**

**199 0.863640 0.868402 0.865872 0.870510 0.039252**

**200 0.846011 0.868471 0.858856 0.849686 0.087595**

**201 0.849372 0.835141 0.842439 0.839206 0.075069**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.695558 0.692375 0.705733 0.692390**

**1 0.700827 0.702649 0.701336 0.694420**

**2 0.693638 0.691473 0.695861 0.688783**

**3 0.665308 0.666084 0.664358 0.657876**

**4 0.619429 0.621425 0.615968 0.610220**

**.. ... ... ... ...**

**197 0.845547 0.832460 0.845149 0.841383**

**198 0.842537 0.829359 0.841743 0.838039**

**199 0.837166 0.825455 0.838910 0.834548**

**200 0.832041 0.827790 0.843656 0.834712**

**201 0.818024 0.812279 0.828237 0.820475**

**MICROSOFT DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.00001 with a 30% reduction callback.

* **Test Loss: 0.12** The value of the Test Loss is 0.638, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE):** 0.0062 which indicates that, on average, the predictions of the model differ only by 0.00894 from the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.008** The RMSE takes into account both the difference between model predictions and actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.93** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.93 suggests that the model explains about 99.2% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.123 at epoch 150**
  + **Best Training Loss: 0.124 at epoch 150**

Immagine che contiene testo, linea, Diagramma, diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.648740 0.674748 0.665816 0.668016 0.206991**

**1 0.632685 0.641160 0.648041 0.651601 0.246201**

**2 0.680053 0.641204 0.658507 0.664833 0.286922**

**3 0.712074 0.679817 0.687700 0.685158 0.327607**

**4 0.715745 0.698241 0.692198 0.713214 0.247972**

**.. ... ... ... ... ...**

**155 0.480893 0.440164 0.464666 0.463714 0.196658**

**156 0.503583 0.479922 0.485598 0.494634 0.180884**

**157 0.518664 0.495350 0.499438 0.515233 0.292948**

**158 0.433127 0.424428 0.444339 0.443525 0.835618**

**159 0.412915 0.423855 0.424401 0.424063 0.357125**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.665754 0.656261 0.649087 0.674750**

**1 0.644831 0.635067 0.630915 0.651110**

**2 0.659988 0.651020 0.648241 0.665474**

**3 0.688405 0.680060 0.677821 0.693313**

**4 0.704400 0.695448 0.688784 0.711287**

**.. ... ... ... ...**

**155 0.466996 0.454408 0.457304 0.471923**

**156 0.495858 0.483208 0.484065 0.501470**

**157 0.508679 0.497614 0.501515 0.512766**

**158 0.435789 0.458412 0.423770 0.423695**

**159 0.425351 0.412899 0.416311 0.423539**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.453251 0.402698 0.436986 0.425291 0.359660**

**1 0.436709 0.435844 0.429721 0.448481 0.220033**

**2 0.419195 0.439547 0.433267 0.431111 0.245424**

**3 0.383503 0.416891 0.414065 0.397963 0.334204**

**4 0.357629 0.375590 0.366967 0.370407 0.314060**

**.. ... ... ... ... ...**

**196 0.829810 0.817120 0.808105 0.848900 0.174424**

**197 0.843211 0.822101 0.815241 0.852856 0.111867**

**198 0.833569 0.829197 0.819696 0.856584 0.090625**

**199 0.827112 0.819500 0.816798 0.851582 0.133029**

**200 0.811544 0.818354 0.805769 0.835258 0.139372**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.431927 0.419590 0.422712 0.429625**

**1 0.442879 0.429924 0.434813 0.447036**

**2 0.435751 0.423569 0.429509 0.440443**

**3 0.408254 0.395655 0.399745 0.407588**

**4 0.373527 0.360332 0.365541 0.372468**

**.. ... ... ... ...**

**196 0.824773 0.818985 0.805250 0.841986**

**197 0.832000 0.821924 0.817024 0.851265**

**198 0.833424 0.822331 0.818081 0.850968**

**199 0.828013 0.819043 0.811948 0.846959**

**200 0.817369 0.808720 0.800714 0.835749**

**AMAZON DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.00001 with a 30% reduction callback.

* **Test Loss: 0.15** The value of the Test Loss is 0.643, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.022** which indicates that, on average, the predictions of the model differ only by 0.022 from the real observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.0039** The RMSE takes into account both the difference between the model predictions and the actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.84** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.838 suggests that the model explains about 83.8% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.1531 at epoch 150**
  + **Best Training Loss: 0.1541 at epoch 150**

Immagine che contiene testo, linea, Diagramma, diagramma

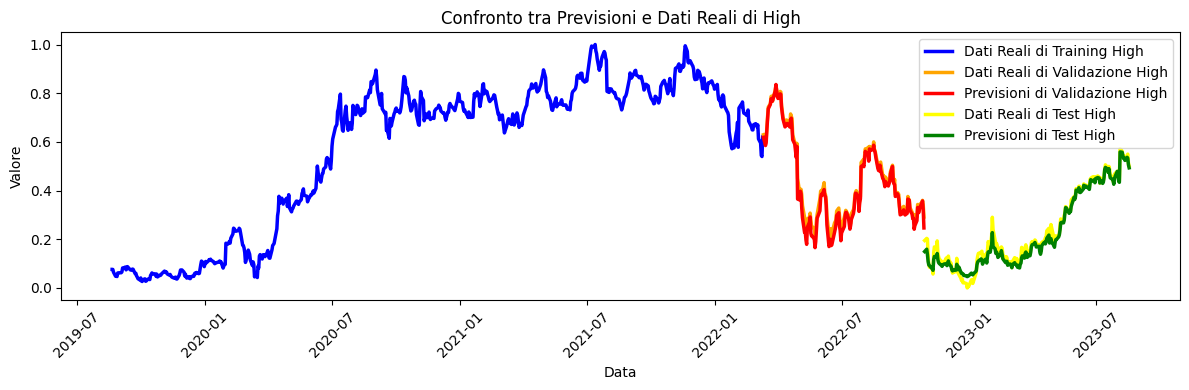
Descrizione generata automaticamente

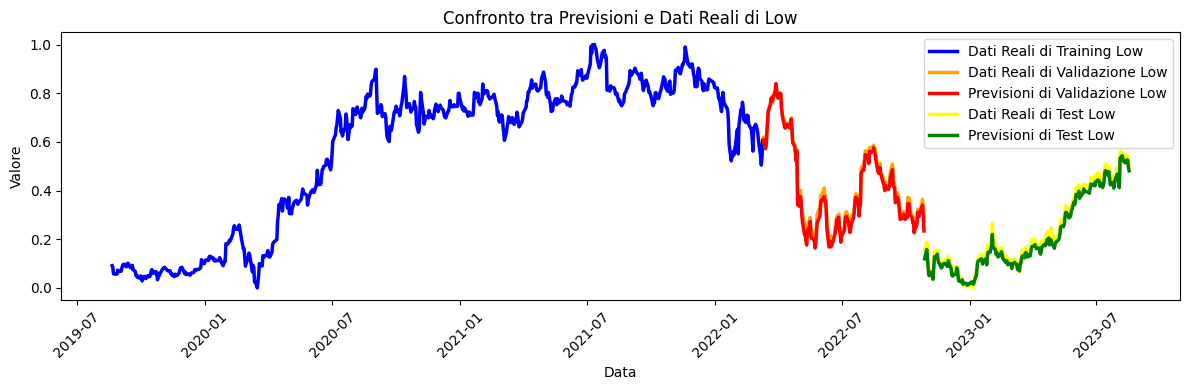
Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente





Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.608155 0.642086 0.629386 0.618783 0.174568**

**1 0.573105 0.607903 0.608230 0.575465 0.192224**

**2 0.625739 0.578119 0.613317 0.586244 0.197324**

**3 0.680512 0.631865 0.662426 0.637949 0.229239**

**4 0.719987 0.671252 0.703772 0.684179 0.188302**

**.. ... ... ... ... ...**

**155 0.357994 0.311198 0.343336 0.320641 0.129492**

**156 0.362767 0.360568 0.350942 0.340633 0.108626**

**157 0.370213 0.357429 0.359785 0.363620 0.113403**

**158 0.323053 0.322708 0.341054 0.323152 0.174235**

**159 0.278185 0.302922 0.291327 0.274957 0.381245**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.622619 0.622078 0.618501 0.608331**

**1 0.590616 0.591864 0.584879 0.571843**

**2 0.601211 0.602113 0.595869 0.583484**

**3 0.654580 0.654824 0.650669 0.640018**

**4 0.693834 0.690960 0.691933 0.685719**

**.. ... ... ... ...**

**155 0.339821 0.346525 0.328714 0.310816**

**156 0.358381 0.363043 0.348184 0.331641**

**157 0.367405 0.371996 0.357303 0.340771**

**158 0.334304 0.344954 0.320446 0.300822**

**159 0.280059 0.328339 0.246310 0.233951**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.206109 0.150626 0.194725 0.157995 0.699671**

**1 0.196849 0.204372 0.203377 0.187743 0.277902**

**2 0.142911 0.208462 0.200620 0.142542 0.462154**

**3 0.098329 0.145014 0.135585 0.103426 0.402206**

**4 0.071408 0.098878 0.095271 0.074548 0.405343**

**.. ... ... ... ... ...**

**196 0.540236 0.526276 0.531025 0.537952 0.085817**

**197 0.560857 0.534837 0.543005 0.545196 0.100513**

**198 0.533172 0.551484 0.549565 0.540174 0.085644**

**199 0.508351 0.524278 0.511438 0.518732 0.081880**

**200 0.497945 0.507822 0.500219 0.504438 0.104617**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.156601 0.289390 0.149328 0.119466**

**1 0.190535 0.240423 0.158969 0.157580**

**2 0.160850 0.244705 0.137707 0.127790**

**3 0.105291 0.197996 0.108213 0.082533**

**4 0.073517 0.176696 0.094160 0.051284**

**.. ... ... ... ...**

**196 0.529158 0.527482 0.523852 0.513744**

**197 0.541811 0.540455 0.536564 0.526224**

**198 0.538441 0.536676 0.533368 0.523506**

**199 0.511234 0.509871 0.505438 0.494677**

**200 0.499966 0.500088 0.493319 0.480974**

**GOOGLE DATASET**

The training is done with 150 eras and a batch-size of 64, learning\_rate=0.00001 with a 30% reduction callback.

* **Test Loss: 0.15** The value of the Test Loss is 0.15, which is relatively low, so the model is able to adapt well to the test data and its predictions are consistent with the real values.
* **Test Mean Absolute Error (MAE): 0.008** which indicates that, on average, the model's predictions differ only by 0.008 from the actual observations, being a very low measure of error and suggests that the model is making very precise predictions.
* **Test Root Mean Squared Error (RMSE): 0.012** The RMSE takes into account both the difference between model predictions and actual values and their dispersion. Such a low RMSE indicates that the model offers very accurate and consistent predictions.
* **R-squared test (R²): 0.93** which is extremely positive because the value of R² measures how much the model can explain the variation in the data compared to a simple reference model.
  + An R² value of 0.93 suggests that the model explains about 85% of the variation in the data, which is very good. In other words, the model has a good forecasting ability.
* **Learning Curves**:
  + **Best Validation Loss: 0.1533 at epoch 150**
  + **Best Training Loss: 0.1534 at epoch 150**

Immagine che contiene testo, linea, Diagramma, diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Let's see a tabular representation of the prediction results on **Validation set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.793278 0.819625 0.818337 0.802531 0.216415**

**1 0.755124 0.785027 0.786125 0.763387 0.256629**

**2 0.784952 0.756437 0.780961 0.764950 0.257047**

**3 0.826125 0.789338 0.813983 0.791489 0.276483**

**4 0.835423 0.812774 0.823812 0.826949 0.187734**

**.. ... ... ... ... ...**

**155 0.497037 0.460778 0.487690 0.479315 0.242704**

**156 0.512260 0.497430 0.502710 0.500180 0.195293**

**157 0.532285 0.509648 0.523007 0.527596 0.252846**

**158 0.428993 0.443613 0.456431 0.442424 0.710615**

**159 0.406312 0.418875 0.422230 0.415512 0.518371**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.811150 0.804115 0.810288 0.811169**

**1 0.773572 0.767544 0.772787 0.775191**

**2 0.773850 0.767397 0.773038 0.775194**

**3 0.807917 0.804254 0.807004 0.811117**

**4 0.827526 0.818710 0.827770 0.828590**

**.. ... ... ... ...**

**155 0.481036 0.470095 0.483284 0.480579**

**156 0.503883 0.489131 0.504296 0.500882**

**157 0.521718 0.510559 0.523476 0.521778**

**158 0.410681 0.402759 0.404382 0.419817**

**159 0.399928 0.389145 0.395941 0.399715**

Let's see a tabular representation of the prediction results on **Test set**:

**Actual Close Actual Open Actual High Actual Low Actual Volume**

**0 0.446975 0.400902 0.439381 0.419775 0.316532**

**1 0.427359 0.433717 0.434205 0.440509 0.252389**

**2 0.384857 0.431799 0.432328 0.400695 0.399335**

**3 0.349813 0.384544 0.382954 0.366223 0.403000**

**4 0.313237 0.338451 0.334747 0.330339 0.457552**

**.. ... ... ... ... ...**

**196 0.790157 0.771184 0.780177 0.786943 0.090857**

**197 0.807117 0.777727 0.795096 0.795410 0.116552**

**198 0.791179 0.795296 0.795908 0.797718 0.086209**

**199 0.779327 0.771971 0.784825 0.784020 0.116796**

**200 0.793120 0.783785 0.800992 0.798031 0.184120**

**Prediction Close Prediction Open Prediction High Prediction Low**

**0 0.424143 0.416997 0.428523 0.424106**

**1 0.432107 0.423110 0.435776 0.431716**

**2 0.404187 0.394165 0.404571 0.401703**

**3 0.361560 0.351242 0.360887 0.359991**

**4 0.313792 0.305933 0.314094 0.319853**

**.. ... ... ... ...**

**196 0.781346 0.763114 0.781858 0.777157**

**197 0.795273 0.778850 0.795140 0.792541**

**198 0.793926 0.776296 0.794678 0.790039**

**199 0.780915 0.764451 0.780985 0.778191**

**200 0.796145 0.786076 0.796158 0.796016**

**CONCLUSIONS**

Through this project, we were able to immerse ourselves in the exciting world of Deep Learning, a branch of artificial intelligence that has revolutionised the way we approach complex data prediction and analysis problems. It was only through relentless efforts and an approach based on experimentation that advanced models capable of learning from large amounts of data, extrapolating intricate patterns and making accurate predictions could be created.

The first step in this project was the theoretical and conceptual study of deep neural networks such as Bi-LSTM and Transformer networks, which were then implemented via python codes on the GoogleColab platform. These networks became the key models for tackling financial time series analysis and prediction problems in Yahoo finance datasets.

They went through the process of information gathering and data preparation, to the optimisation of the models, both in terms of architecture and parameters. Different normalisation modes, activation functions, optimisers and regularisations were experimented with to improve the models' ability to generalise to new and unknown data.

Finally, ensemble learning concepts were applied, combining the results of multiple models to obtain more robust and accurate predictions. This made it possible to create hybrid models that make the best of different network architectures and deep learning approaches.

Some future developments related to this project could include the use of a grid search, with different hyperparameters, for further improvements of the models used. Furthermore, the application of the models created in a supervised learning context on an unsupervised learning task could be explored. For example, it might be interesting to experiment with the generation of new dates, for which the models will need to be able to accurately predict feature values.

This project was not only a journey into the universe of Deep Learning, but also an opportunity to tackle real-world challenges through artificial intelligence.