



PyData Venice

Stock Market Forecasting with Conformal Prediction and Stock Analysis with AI Agent Recommendations

Claudio Giorgio Giancaterino
30/10/2025

MySelf:

Actuary during the day &
AI Scientist in the free time

Reach Me:

MyLinks



Agenda

Motivations

Uncertainty Quantification & Conformal Prediction

NIXTLA Conformal Prediction

AI Agent

Pyautogen AI Agent

Web App in Action

Motivations

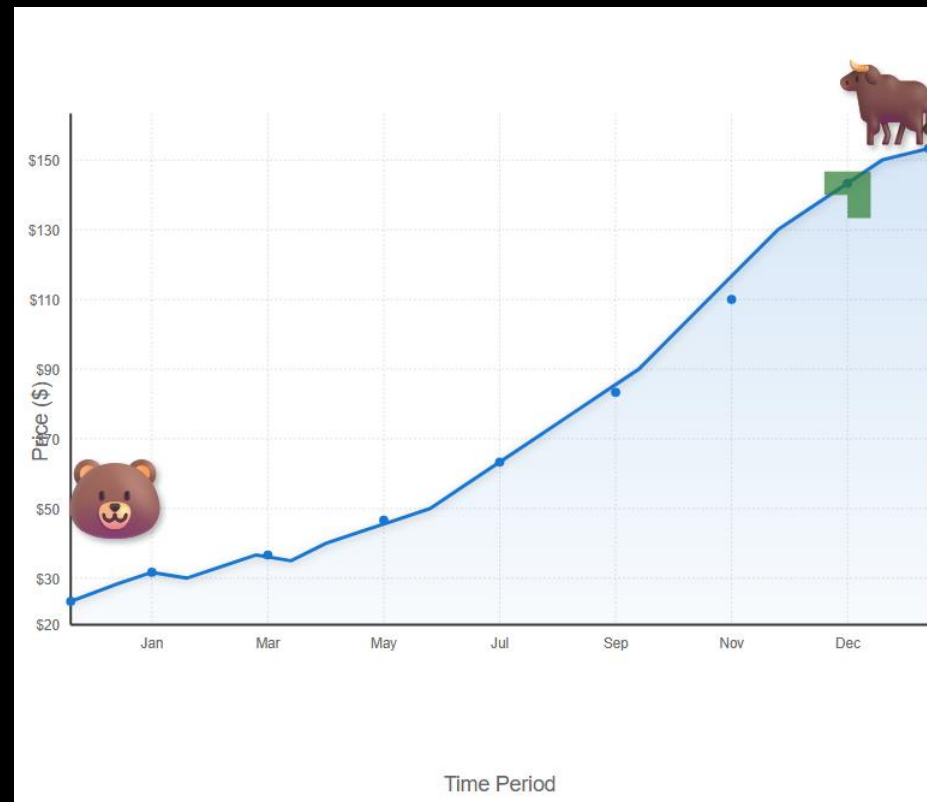
- Build a Predictive AI use case
- Experiment Conformal Prediction with stocks
- Try to merge Predictive AI with the power of Generative AI

Google Banana



Why does uncertainty quantification matter?

Stock Investment Risk



<https://deepai.org/machine-learning-model/text2img>

Why does uncertainty quantification matter?

- Uncertainty quantification is a crucial aspect of predictive modelling. Traditional predictions typically produce a single-point output for a given input because they rely on models that provide only the most likely outcome. These models either cannot capture the full complexity of the real world or are based on incomplete data.
- Uncertainty can be broken into:
 - **Aleatoric Uncertainty.** This type of uncertainty arises from inherent randomness in the data. It is also known as statistical or irreducible uncertainty. In machine learning, aleatoric uncertainty can result from noise in the data or measurement errors.
 - **Epistemic Uncertainty.** This type of uncertainty arises from a lack of knowledge or information. It is also known as systematic or reducible uncertainty. Epistemic uncertainty can be reduced by gathering more data or improving the model.

Why Conformal Prediction ?



Conformal prediction is a powerful machine learning framework used to evaluate the uncertainty of predictions. It turns point predictions into prediction sets/intervals.

Some advantages of using conformal prediction for uncertainty quantification:

- distribution-free;**
- guaranteed coverage;**
- model-agnostic;**
- easy implementation**

Conformal Prediction Properties



Exchangeability

ensures that predictions are consistent regardless of data order. For time series data, some mitigation approaches are required.



Validity

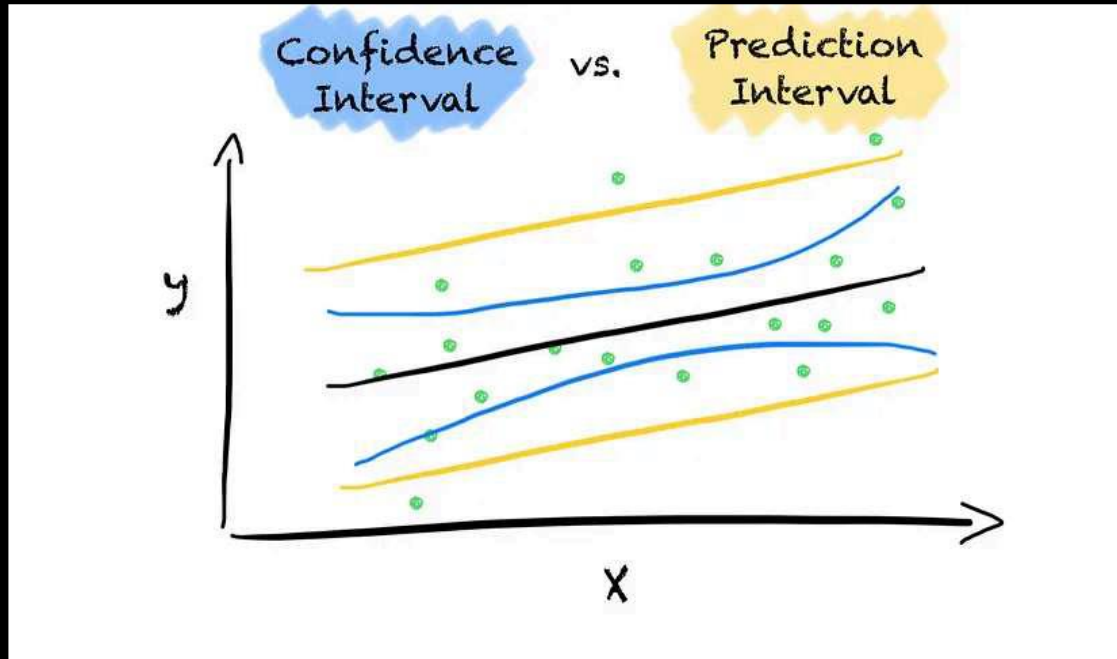
ensures that predictions are reliable and meet predefined coverage levels, making it crucial for applications where trust in predictions is relevant.



Efficiency

focuses on producing accurate and concise predictions, which is vital for practical implementation and resource management.

Confidence Intervals vs Prediction Intervals

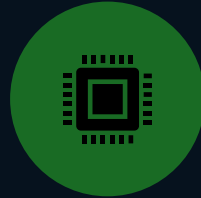


<https://medium.com/data-science/confidence-interval-vs-prediction-interval-a6b0c4816a92>

- The Confidence Interval shows the uncertainty of a population parameter, such as the mean.
- The Prediction Interval shows the uncertainty of a specific value.
- The Confidence Interval focuses on past or current events.
- The Prediction Interval focuses on future events.
- A Prediction Interval is generally wider than a Confidence Interval, because the Prediction Interval must account for the additional uncertainty of a single observation compared to the mean.

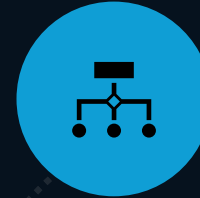


Full Conformal Prediction (Transductive) leverages the entire dataset for training and requires model retraining for each new prediction. This approach ensures that each prediction is made with the most up-to-date information. It's the most general and computationally intensive approach.



Split Conformal Prediction

(Inductive) splits the data into training and calibration sets. The model is trained once on the training set, and the calibration set is used to adjust the prediction intervals. It's the computationally efficient variant.

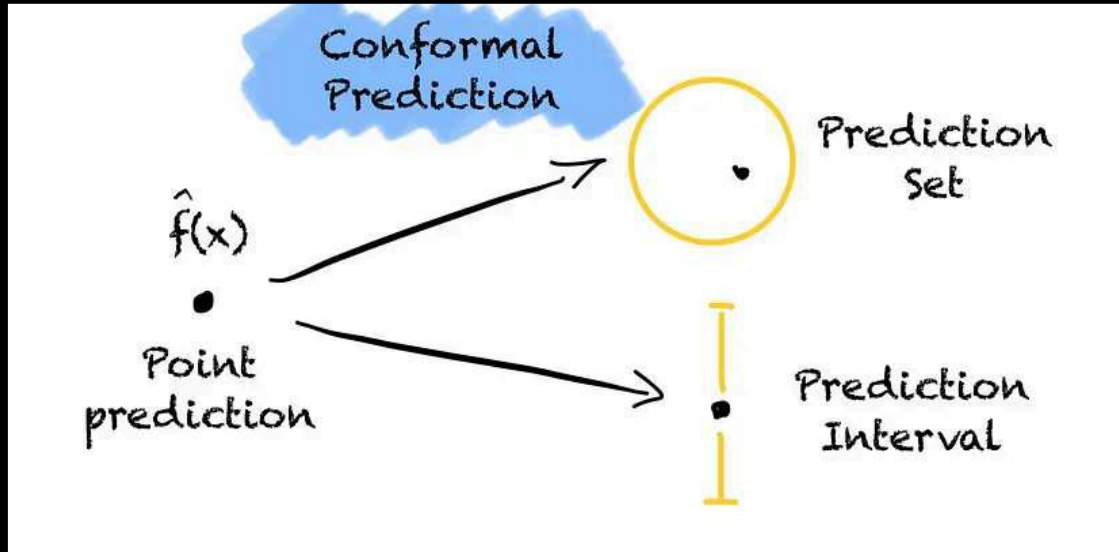


Cross-Conformal prediction

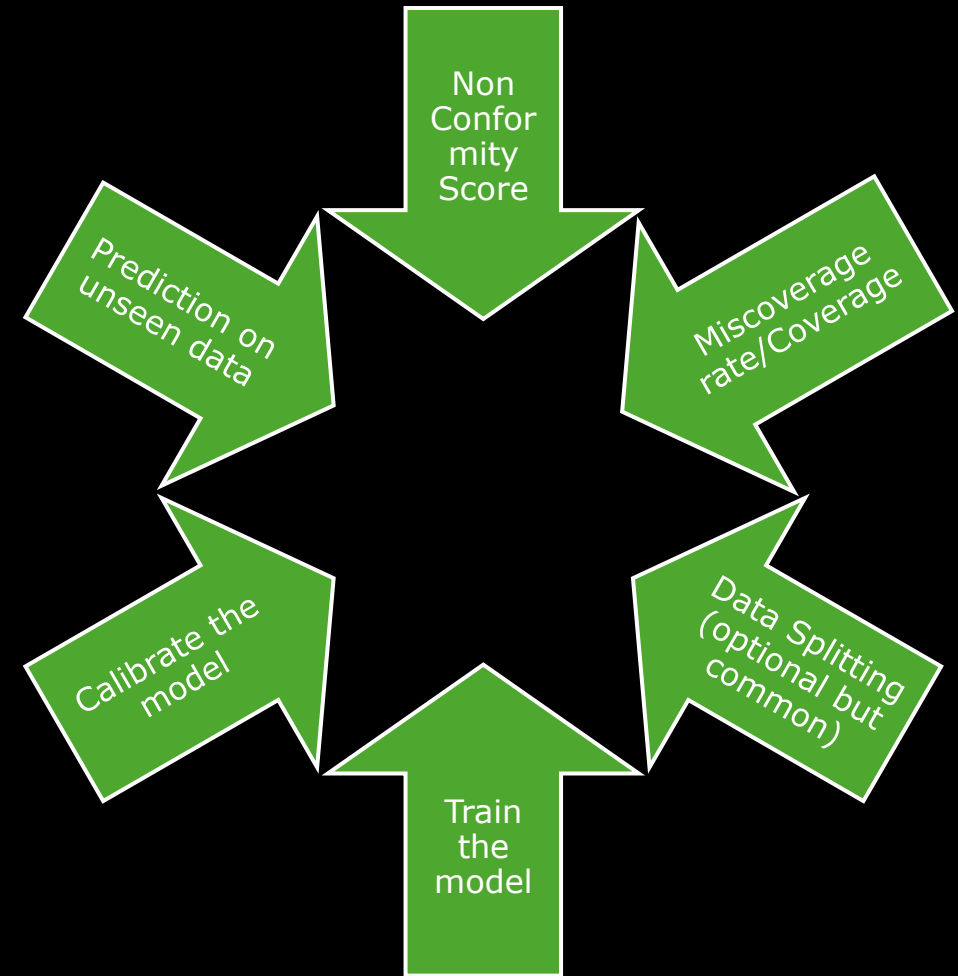
is a resampling-based method that generalises the split conformal prediction. Data is split into K folds; for each fold, the model is trained on the remaining $K-1$ folds, and the prediction interval is computed on the left-out fold. The final prediction interval is built by aggregating results across all partitions.

Conformal Prediction Methods

Conformal Prediction as a Recipe



<https://medium.com/data-science/all-you-need-is-conformal-prediction-726f18920241>



Conformal Prediction as a Recipe: numerical example on Time Series

First Goal

Build a **90% prediction interval** for **time series forecasting** using **conformal prediction**.

Assumptions

-15 consecutive timestamps of data.

-Predict the next value in the series using some forecasting model.

Split dataset:

-**First 10 timestamps** as the **training set**

-**Next 5 timestamps** as the **calibration set**

The conformal prediction goal is **90% coverage**, so $\alpha = 0,1$.

Split Time Series

Time 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Value 80 82 83 92 94 85 88 94 98 96 99 91 96 90 89

- Train a model on timestamps 1–10
- Predict timestamps 11–15 using the model trained on 1-10
- Compute **non-conformity scores** for 11-15 (calibration) -> $s = |y - \hat{y}|$
- Use those scores to form intervals for the future timestamp (16)

Conformal Prediction as a Recipe: numerical example on Time Series

✓ Train the Forecasting Model and get errors:

Time	11	12	13	14	15
True outcome	99	91	96	90	89
Predicted outcome	97	95	94	91	90
Non-conformity scores	2	4	2	1	1

✓ Sort the non-conformity scores (in ascending order): [1,1,2,2,4]

✓ Compute the quantile index (with finite sample correction) for the n samples

$$k = \min(n, [(1 - \alpha)(n + 1)]) = [0,9 \times 6] = [5,4] = 5$$

Set the threshold value from the sorted non-conformity scores $\Rightarrow \hat{q}=5\text{th}=4$

✓ Build the prediction Interval

Given a prediction at the 16 timestamps $\hat{y}=92$

The 90% conformal prediction interval is: $[\hat{y} - \hat{q}, \hat{y} + \hat{q}] = [92 - 4, 92 + 4] = [88, 96]$

Conformal Prediction as a Recipe: numerical example on Time Series

Second Goal

Build a **90% prediction interval** for timestamp 17 using a rolling window approach. This preserves **temporal order** and allows the non-conformity scores to reflect the most recent data distribution, partially mitigating the **non-exchangeability issue** that violates standard conformal prediction assumptions.

Split Time Series

Time	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Value	82	83	92	94	85	88	94	98	96	99	91	96	90	89	97

- Train a model on timestamps 2–11
- Predict timestamps 12–16 using the model trained on 2-11
- Compute **non-conformity scores** for 12-16 (calibration) $\rightarrow s = |y - \hat{y}|$
- Use those scores to form intervals for the future timestamp (17)

Conformal Prediction as a Recipe: numerical example on Time Series

✓ Train the Forecasting Model and get errors

Time	12	13	14	15	16
True outcome	91	96	90	89	97
Predicted outcome	95	94	91	90	92
Non-conformity scores	4	2	1	1	5

✓ Sort the non-conformity scores (in ascending order): [1,1,2,4,5]

✓ Compute the quantile index (with finite sample correction) for the n samples

$$k = \min(n, [(1 - \alpha)(n + 1)]) = [0,9 \times 6] = [5,4] = 5$$

Set the threshold value from the sorted non-conformity scores $\Rightarrow \hat{q}=5\text{th}=5$

✓ Build the prediction Interval

Given a prediction at the 17 timestamps $\hat{y}=93$

The 90% conformal prediction interval is: $[\hat{y} - \hat{q}, \hat{y} + \hat{q}] = [93 - 5, 93 + 5] = [88, 98]$

Conformal Prediction with NIXTLA



Open Source Time Series Ecosystem

Stars 6.1k

StatsForecast

Lightning fast forecasting with statistical and econometric models.

Github

MLForecast

Scalable machine learning for time series forecasting.

Github

NeuralForecast

Scalable and user friendly neural forecasting algorithms for time series data.

Github

Hierarchical Forecast

Probabilistic Hierarchical forecasting with statistical and econometric methods.

Github

TS features

Calculates various features from time series data. Python implementation of the R package *tsfeatures*.

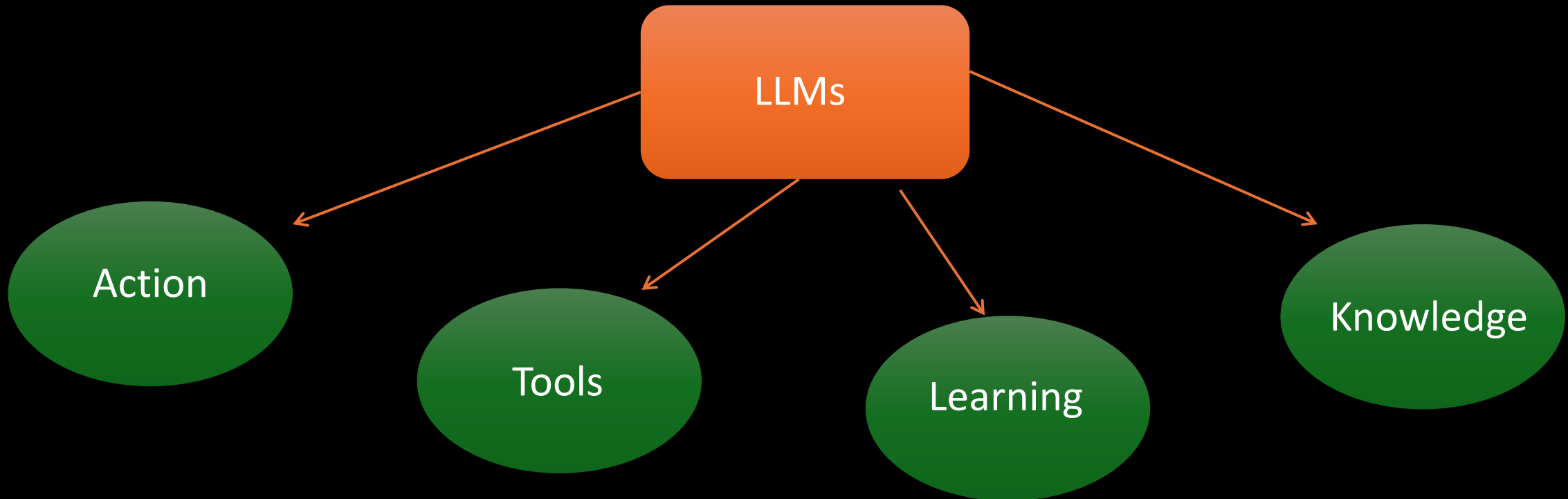
Github

<https://www.nixtla.io/open-source>

- ❑ **Nixtla** is an open-source framework dedicated to time series forecasting, and it is helpful for low-code.
- ❑ Organized into the following libraries:
 - StatsForecast;
 - MLForecast;
 - NeuralForecast;
 - TimeGPT
- ❑ Conformal Prediction Approaches:
 - Conformal Distribution;
 - Conformal Error
- ❑ Nixtla's Approach to Non-Exchangeable Data using Rolling Window Cross-Validation for Interval Calibration

What are AI Agents?

An AI agent is an application designed to achieve a goal by observing the world and acting upon it using the available tools. These agents are autonomous and can act independently of human intervention when given proper goals. In the context of Generative AI, an agent is a program that enables Large Language Models (LLMs) to perform actions by extending their capabilities through access to tools, knowledge, and learning from experience.



Reasoning

Collaborating

Why AI Agents?

-> Autonomously Perform Tasks

-> Adapt and Learn

-> Increase Productivity and Efficiency

-> Enable Personalisation and Scalability

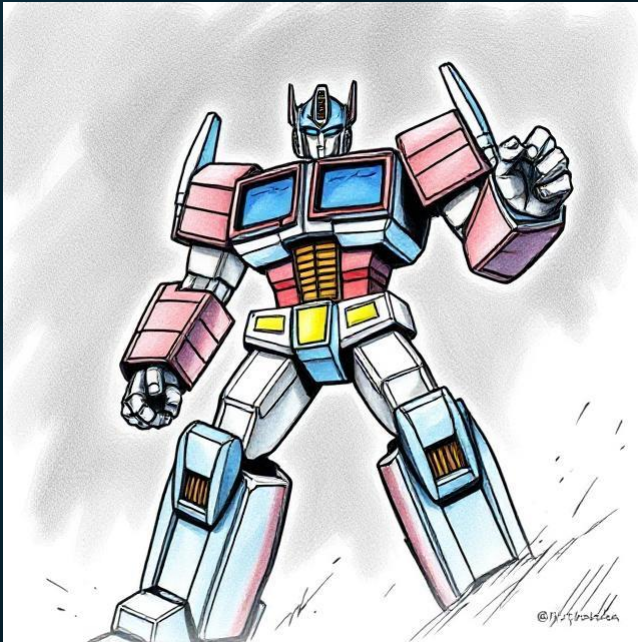
Acting

Observing

Self-Refining

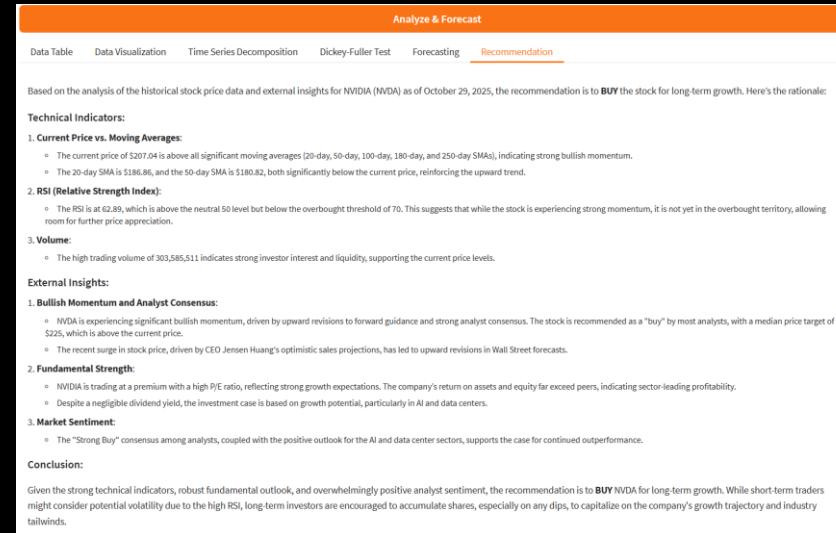
Planning

pyautogen framework for AI Agents development

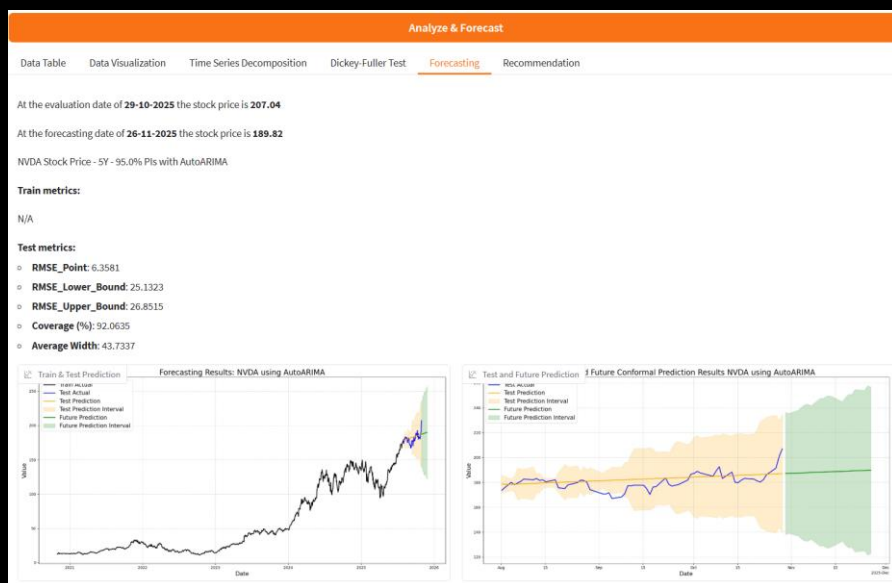


- AutoGen (pyautogen), now as AG2, is an open-source programming framework originally developed by Microsoft. It is designed for building AI agents and sophisticated cooperation among multiple agents in order to solve complex tasks.

Build the App






Hugging Face app



Conclusions

- **What I've learned**

-  NIXTLA for time series is a good open-source framework projected for use cases with low code and the opportunity for scalable solutions.
-  AI Agents employed as recommendation systems are powerful in explaining your data and in managing external resources.
-  Hugging Face Spaces are GitHub repositories that are easy to use and can be used as your portfolio.

References

[A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification](#)

[Theoretical Foundations of Conformal Prediction](#)

[A Tutorial on Conformal Prediction](#)

[Conformal Prediction: A Unified Review of Theory and New Challenges](#)

[Conformal Prediction Guide](#)

[Practical Guide to applied Conformal Prediction in Python](#)

[Introduction to Conformal Prediction in Python](#)

[Awesome Conformal Prediction](#)

[Forecasting: Principles and Practice, the Pythonic Way](#)

[pyautogen · PyPI](#)

[Gradio Documentation](#)

[Introduction - TimeGPT Foundational model for time series forecasting and anomaly detection](#)

[StatsForecast ⚡ - Nixtla](#)

[mlforecast - Nixtla](#)

[About NeuralForecast - Nixtla](#)

[Notebooks](#)

[stock-market-forecasting-with-nixtla-conformal-prediction](#)

Thank you

?

?

?

?

?

?

?

Keep in touch:

- **Linkedin**
- **Newsletter**
- **Medium**
- **Website**