Svedberg Open
DISSEMINATIONOFKNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/



Research Paper Open Access

An Automated Meta- Forecasting Framework For Intelligent Model Selection in Retail Time Series Analysis

Riddhi J. Kotak[1], Jinal H. Tailor[2], Amit Lathigara[3], Krunal Vaghela[4]

[1]Research Scholar, PhD-Computer Engineering/IT, Gujarat Technological University, Ahmedabad. Email: kotakriddhi@gmail.com
[2]MCA,S. S. Agrawal Institute of Management & Technology, Navsari, India. Email: jinal.tailorssa@gmail.com
[3]RK University, Rajkot, India. Email: amit.lathigara@rku.ac.in
[4]Engineering & Technology, Marwadi University, Rajkot, India. Email: krunal.vaghela@marwadieducation.edu.in

Abstract
The inability to predict the suitability of a forecasting model for each and every sales series makes retail sales forecasting difficult. This research introduces an intelligent meta-forecasting framework that integrates data-driven model recommendation, quantitative forecasting, retrieval-augmented generation (RAG) and large language model (LLM) decision support. Experiments were carried out with 10,194 retail transactions made between 2019-2022. The results showed that there are 17 sub-category sales series, which led to the creation of 68 forecasting tasks for the evaluation of the meta-learning. When tested on unseen retail series, the Random Forest meta-classifier outperformed the other models with a model-selection accuracy of 40.0% and a macro-F1 score of 0.273. The mean test RMSE was 259.349, which was significantly better than random selection (RMSE 316.931; regret 68.997), but was not as good as a global baseline of SGD (RMSE 250.967; regret 3.032). A prototype RAG–LLM system is also instantiated with semantic retrieval, model recommendation and interpretation of forecasts through the conversation. The results show the suitability of a forecasting model to be heterogeneous and also highlight the limitation of cross-series generalization in automated meta-learning.

Index Terms: Meta-Learning, Retail Demand Forecasting, Time Series Modeling, Automated Machine Learning, Interpretable AI, Model Evaluation.

1. Introduction

The demand forecast is an essential part of the supply chain and inventory management system in today's world. Forecasts can help retailers make better-informed decisions about their inventory, minimize waste, and deliver superior customer service. Retail time series are, however, complex as they have to account for the changing trends, seasonality, promotions, variations in demand, and other external influences. Prophet offers a scalable forecasting framework that explicitly models the trend and seasonal components, and has been extensively used for forecasting business time-series (Taylor and Letham, 2018).

Complex nonlinear relationships within retail data can be hard to capture using traditional forecasting methods based on statistics. Thus, machine-learning and deep-learning techniques have become more and more popular choices for forecasting retail sales. In recent years, hybrid deep-learning architectures of convolutional neural networks and Long Short-Term Memory networks have been proven to be effective for the modelling of both exogenous and temporal information in retail sales data (de Castro Moraes et al., 2024). However, no individual forecasting model is likely to be best for all the time series, which might be quite different. Meta-learning is thus a valuable tool for learning relationships between time-series characteristics and performance of forecasting models. Ma and Fildes (2021) investigated meta-learning specifically for retail sales forecasting and demonstrated how learned time-series representations can be related to a pool of base forecasting methods. A more recent work is that of Talkhi et al. (2024) which showed how the time-series characteristics and machine-learning classifiers can be used as meta-learners for recommending appropriate forecasting models.

Explainability is also crucial when AI forecasting systems are applied in real life decision support. To address the challenges posed by the characteristics of various data types, Chen et al. (2024) proposed an explainable

sales-forecasting framework that combines various data types: neural forecasting, textual information, multiple regression, and SHAP-based interpretation. These advances are a motivating factor for proposing the framework that integrates automated model selection, based on meta-features, together with contextual retrieval and natural-language explanation.

The main contribution of this study is an automated meta-forecasting framework that considers the problem of model selection as a meta-learning problem instead of the traditional exhaustive model evaluation. Under the proposed framework, the relationships between the characteristics of a new time series and forecasting-model performance would be learned once and for all, and a new time series would be characterized by a meta-feature vector which uses these relationships to predict which forecasting algorithm is most likely to be appropriate for that time series.

The framework separates out the selection of the models from contextual explanation. First, the statistical and structural meta-features, such as trend strength, seasonal strength, autocorrelation, entropy, volatility and complexity measures are extracted from the historical data that is available prior to the forecast origin. Second, the meta-learner applies this knowledge to a candidate forecasting model to map these characteristics to the candidate forecasting model. Third: A numerical forecast is produced by the chosen forecasting model. Lastly, the RAG and LLM parts gather pertinent historical and field-specific data and translate the evidence and forecast from the model into a natural-language explanation in context.

The LLM is thus not considered to be a replacement for the forecasting algorithm or a non-constrained way of selecting an arbitrary forecasting model. The contextual augmentation and explanation are done by RAG and the LLM and the principal model-selection decision is generated by the meta-learning mechanism. This separation allows for better reproducibility as quantitative forecasting is still under the control of explicit forecasting algorithms, which are evaluated.

The key contributions of this work are: (1) formulation of retail forecasting-model selection as a meta-learning task based on statistical and structural time-series characteristics; (2) construction of an automated pipeline for meta-feature extraction, model recommendation, forecasting and interpretation of the resultant forecasting and model-selection results; (3) evaluation of heterogeneous statistical, machine-learning, and deep-learning candidate models; (4) incorporation of RAG and an LLM as an explanation layer to the forecasting and model-selection results; and (5) empirical evaluation of forecasting accuracy and model-selection quality for heterogeneous retail time series.

Scientific Novelty and Contribution: The current work is based on an investigation of automated retail forecasting with complementary model-selection mechanisms based on data and context. The meta-learning part represents the recommendation of forecasting models as a classification task, where both statistical and temporal features of the historical retail series are transformed to empirical appropriate forecasting algorithms.

This is distinct from traditional single model forecasting and extensive model testing, as past relationships between time-series attributes and model accuracy are re-used for new forecasting problems. At the same time, an implemented RAG–LLM pipeline processes the data to conduct semantic retrieval and constrained contextual recommendation. Conventional forecasting algorithms will continue to be used to make numerical predictions, which will allow for evaluation of forecasting accuracy and the quality of the predictions through these algorithms to be assessed separately.

2. Background & Literature Review

Retail time series forecasting has advanced greatly in the last decade, shifting its focus on classical statistical methods, to more advanced machine learning, deep learning, and meta-learning-based methods. This section will give a critical review of the recent literature (2026-2021) with respect to the methodologies used and datasets, as well as significant contributions to the topic of automated model selection and forecasting accuracy.

TABLE 1. COMPARATIVE STUDY OF LITERATURE REVIEW

Author Name	Paper Published	year	Dataset	Methodology
Taylor and Letham	2018	Business time series	Prophet	Trend and seasonality modelling

Ma and Fildes	2021	Retail sales	Meta-learning	Retail forecasting using meta-learning
de Castro Moraes et al.	2024	Retail sales	Hybrid CNN-LSTM	Deep-learning retail forecasting
Talkhi et al.	2024	Multiple time series	Meta-learning + ML	Automated forecasting-model recommendation
Chen et al.	2024	Sales/e-commerce	WaveNet + SHAP	Explainable AI-based sales forecasting
Taylor and Letham	2018	Business time series	Prophet	Trend and seasonality modelling

Ma and Fildes (2021) studied meta-learning in the context of retail sales forecasting and found that information gained from retail time series can be used to forecast other heterogeneous series. They examine how relevant meta-learning is to situations in retail markets where the optimal forecasting method might change depending on the nature of the series. Talkhi et al. (2024) also showed that the machine-learning meta-learners are able to recommend an accurate forecasting model based on time-series characteristics. The provided studies offer a methodological basis to consider forecasting-model selection as a meta-learning problem.

More recently, research has focused on explainability and deep learning in the field of sales forecasting, with de Castro Moraes et al. (2024) exploring how a hybrid CNN-LSTM model could be used to forecast retail sales, highlighting the promise of deep-learning techniques in capturing the nuances of temporal relationships. In the context of sales forecasting, Chen et al. (2024) integrated EAI by providing explanations through SHAP. But most of the studies aim at prediction or interpretation but do not consider the problem of choice of forecasting models, meta-features recommendation, contextual retrieval and natural-language explanation and are not integrated in a unified framework.

2.1 Research Gap and Motivation

The existing research in retail forecasting shows impressive results when adopting statistical, machine learning and deep learning methods, but none of the forecasting algorithms is able to achieve best performance on all the diversified characteristics of the time-series. When the trend and seasonal patterns are structured, statistical methods may be useful; when there are enough observations and explanatory variables and the relationships are nonlinear, machine-learning and deep-learning methods might be more useful. Therefore, choosing the suitable forecasting algorithm is still a challenging issue.

While AutoML eliminates manual experimentation, typical automated selection typically involves retraining and re-evaluating the candidate models for different forecasting tasks. To overcome this, a new approach known as 'Meta-learning' is suggested in which the relations between the characteristics of the datasets and the performance of the previous models are learned. However, there are three issues that are still applicable to both actual retail forecasting.

2.1.1. The suitability of these forecasting models to the different interpretable characteristics of time series needs to be systematically assessed in different and heterogeneous retail series.

2.1.2. Performance of model selection should be judged apart from the accuracy of the final forecast. In addition to having a low RMSE or MAE a forecasting tool should show that the meta-learner chooses or closely approximates the empirically best candidate model.

2.1.3. Third, the selection of the numerical models and the explanation of the context should be clearly distinguished. The natural-language explanations produced by an LLM should be based on model-selection evidence and retrieved knowledge—don't get rid of the quantitative forecasting procedure.

In this study, the question is whether statistical and structural meta-features can be used to learn the generalizable mapping between the characteristics of time-series data found in retail stores and the performance of candidate-models, as well as whether natural-language explanations can be added on top of these features without degrading the reproducibility of the forecasting decision made by the model.

3. Proposed system Overview

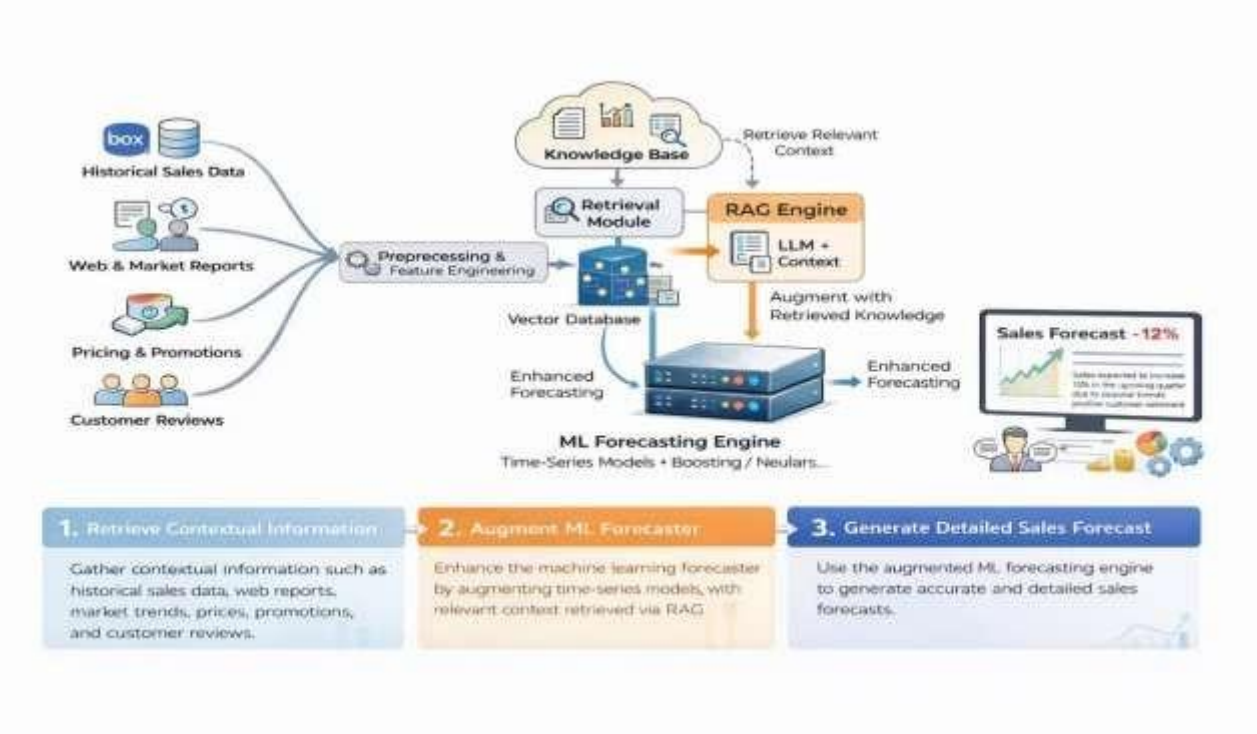


Fig.1 Proposed system Overview

The framework proposed combines Retrieval-Augmented Generation (RAG) with traditional machine learning models to improve the accuracy and interpretability of retail sales forecasts. The architecture starts with the group of heterogeneous sources of data, such historical sales data, market reports, pricing strategies, promotional activities, and customer reviews.

An embeddings-based vector database is then built by storing embeddings of processed data, allowing similarity-based retrieval to be easily performed. The retrieval module retrieves the appropriate contextual knowledge information in the knowledge base depending on query input. This contextual memory is fed to the RAG engine, where a Large Language Model (LLM) takes the contextual memory and input features to produce enriched insights.

The augmented information is then inputted into the machine learning forecasting engine, which can take the form of time-series models, ensemble methods, or deep learning. The forecasting model generates better predictions that are explainable, due to the incorporation of external contextual knowledge.

The general procedure can be divided into three main steps: Context Retrieval - retrieving the externally relevant knowledge.

Model Augmentation – improving ML models with RAG.

Generation of Forecasts- generating detailed and context-aware sales forecasts.

It is a hybrid method, which combats the shortcomings of the traditional forecasting systems by integrating knowledge-based reasoning with data-based learning, thus enhancing the forecasting performance in dynamic retailing settings.

1. Sales Time Series S = {s1, s2, ..., sT}

Where s t is the sales in the t th period and T is total observations.

2. Contextual Features

X_t = {x_t^(1), x_t^(2), ..., x_t^(m)}

Where m is number of external features such as price, promotion, etc.

3. Forecasting Objective

$\hat{s}(T+h) = f(S(1:T), X(1:T), C_T)$
Forecast future sales based on past data and contextual information.

4. Embedding Function $v_i = \varphi(d_i)$
Document d i is transformed into v i.

5. Similarity Measure
$sim(q_T, v_i) = (q_T \cdot v_i) / (||q_T|| ||v_i||)$
Relevant documents were found by using cosine similarity.

6. Retrieval
$R_T = TopK(sim(q_T, v_i))$
Relevant documents of top-k are obtained.

7. RAG Generation
$P(C_T | q_T) = \Sigma P(C_T | q_T, d_j) P(d_j | q_T)$ Integrates knowledge that has been retrieved and query.

8. Forecast Model $\hat{s}(T+h) = f(S, X, C_T; \theta)$
Forecast relies on past experience, characteristics, and background.

9. Neural Representation
$h_T = \sigma(Ws S + Wx X + Wc C_T + b)$ Combined representation of all the inputs.

10. Output Layer $\hat{s} = Wo h_T + bo$
End predicted value of sales.

11. Loss Function (MSE) $L = (1/n) \Sigma (s_i - \hat{s}_i)^2$ Minimizes prediction error.

4. Data Preparation

The proposed framework uses both the structured (historical sales, pricing, promotions) and the unstructured data (customer reviews, market information) to enhance the accuracy of the forecast. Data preprocessing involves processing of missing values, outlier data, normalizing and encoding of categorical data in order to achieve consistency in data.

The data set consists of 10,194 transaction records over a period of 3 January 2019 to 30 December 2022 from the Superstore retail dataset, which includes 21 original attributes. There are 5,111 different orders, 1,862 product IDs, 1,849 product names, three categories, 17 sub-categories, four geographical regions, 542 cities and 804 customers. Attributes include order and shipping information, customer and geographical characteristics, product information, sales, quantity, discount and profit. Sales was chosen as the most important forecasting variable. For the evaluation of the meta-learning, transactions were consolidated into the daily sales series at the sub-category level, resulting in 17 heterogeneous series of retail sales of 1,458 calendar days each.

The feature engineering is done by creating lag features, rolling statistics, and calendar-related features to represent the temporal patterns and seasonality. Textual data that is unstructured is processed via techniques of Natural Language Processing where documents are converted into embeddings using:
$v_i = \varphi(d_i)$
These embeddings are stored in a vector database to be easily accessed in the RAG framework. Lastly, the features which are processed are then combined into a single dataset:
$D_t = \{s_t, X_t\}$
In which s t = sales and X t = auxiliary features. A time-based method is used to divide the dataset into training and testing sets to maintain the time dependencies of the dataset.

5. Meta - Feature Extraction and Meta - Learning Model

In this section, the key aspect of the proposed framework is described, which allows the intelligent and automated choice of a model with the help of meta-feature extraction and a meta-learning model. The aim is

to convert raw time series data into descriptive statistical properties (meta-features) and apply them to produce predictions of the most appropriate forecasting model.

5.1 Meta–Feature Extraction

Meta-features are statistical and structural features that characterize the behavior of a time series dataset. These characteristics are important in the detection of patterns like trend, seasonality and noise which have a direct impact on model performance. The statistical and structural features of time series behavior are referred to as meta-features.

Statistical Features:

- Mean, variance, skewness, kurtosis
- Entropy: H(X) = - Σ p(x) log p(x)

Trend Features:

Trend shows long-range trend in the data.
The trend strength = 1 - Var(residual)/ Var(trend + residual) Seizes growing or declining demand trends.
Helps determine that models such as Prophet are suitable.

Seasonality Features:

Seasonality is a repetition of some patterns at a specific time. Seasonal strength Frequency detection (weekly, monthly)

Seasonality Strength = 1 - Var(residual) / Var (seasonal + residual)

Autocorrelation Features:

Determines dependence between observations. ACF (Auto-Correlation Function)
PACF (Partial Auto-Correlation Function) ACF(k) = Σ (x_t - μ)(x_{t-k} - μ) / Σ (x_t - μ)^2

Complexity Features:
Approximate entropy Hurst exponent Nonlinearity tests

These characteristics can be used to decide whether deep learning models (e.g., LSTM) are needed.

- Entropy, Hurst exponent, nonlinearity.

Meta-feature vector:
φ(D) = [f1, f2, ..., fn]

Meta-features are statistical and structural features that characterize the behavior of a time series dataset. These characteristics are important in the detection of patterns like trend, seasonality and noise which have a direct impact on model performance. The statistical and structural features of time series behaviour are referred to as meta-features.
A meta-feature vector ϕ(Di) was derived only from historical observations that were available prior to the forecast origin for forecasting task (Di). Evaluated features were distribution, variability, sparsity, trend and seasonal behavior. The best model was selected for each candidate forecasting model independently over the validation interval, and the minimum-RMSE model was given as:

$m_i^{\wedge} *= arg\ min\top(m \in M)\ R\ \ MSE(m,$ this resulting

meta-dataset was:

$D_{meta} = \{(\phi(D_i), m^*)\}^{68}$
i i=1
Ridge Regression performed best at 68 forecasting tasks, with a best performance at 29 of those tasks, SGD Regression performed best at 26 of those forecasting tasks, and Random Forest Regression performed best at

13 of those forecasting tasks.

5.2 Meta–Learning Model

The current setup involves formulating the problem of automated model selection as a multiclass classification problem. Let \(D_i\) be the \(i^{th}\) retail time series and \(\phi(D_i)\) its meta-feature vector. A set of candidate forecasting algorithms, \(M=\{m_1,m_2,\ldots,m_K\}\), are independently trained and tested on a set of validation data that is temporally ordered. Each time series is associated with a candidate model with the lowest RMSE in the validation data and this is labeled as the optimal-model label:
yi∗=argm∈MminRMSE(m,Di) The resulting meta-dataset is:
D_{meta}=\{(\phi(D_i),y_i^*)\}_{i=1}^{N}

A meta-classifier \(g(\cdot)\) is then trained so that it learns: g:\phi(D_i)\rightarrow y_i^*
Meta-features are computed from historical observations for a time series that will not be used for forecasting (the \(D_{new}\) series), and the trained meta-classifier predicts:

hat m=g(\phi(D_{new}))

Where m is the recommended forecasting model.

This formulation is not to be confused with stacking. Stacking is a form of meta-learning that uses multiple base learners to generate predictions that are integrated to form the final prediction. The meta-learner is responsible to recommend algorithm in the proposed framework, and it doesn't fuse the prediction by the candidate models.

5.3 Prevention of Temporal Information Leakage

Special effort was made to avoid temporal information leakage. All processing parameters, meta-parameters, model parameters, and model selection decisions were only made using observations that were available prior to the date of the forecast's origin. Meta-features such as trend, seasonality, autocorrelation, entropy, scaling parameters and other features of the complete time series were not calculated before evaluation.
Meta-features were computed for each origin \(t\) of the forecasting data, using the historical data window \(D_{1:t}\). The performance labels of the candidate models were calculated based on the specified validation period only, while the training of the candidate models was based on observations before the specified validation period. During the evaluation of the different models with the test intervals as well as the training of the meta-learner, there was no knowledge of the final test interval.
Chronological splitting was used instead of random train-test splitting. In cases where rolling-origin evaluation was used, the training window was extended in a step-by-step fashion and the next period of the forecasting horizon was not viewed. This removes the possibility of information that was never available at the time of the forecast being used to either boost the performance of the meta-features or recommend an algorithm.

6. Candidate Forecasting Models and Meta-Learner

The proposed meta-forecasting framework clearly separates the candidate forecasting models from the meta-learner responsible for intelligent model selection. Candidate models generate numerical sales forecasts, whereas the meta-learner analyzes the statistical and temporal characteristics of a retail time series and recommends the forecasting model expected to provide the best performance. This formulation differs from stacking because the meta-learner does not combine predictions from multiple base models; instead, it performs algorithm recommendation as a multiclass classification task.

6.1 Candidate Forecasting Models

A heterogeneous candidate-model pool comprising statistical, machine-learning, ensemble-learning, and deep-learning approaches is considered to accommodate different retail time-series characteristics.
ARIMA/SARIMA: ARIMA is included as a classical statistical model for capturing autoregressive and moving-average dependencies, while SARIMA extends it by modelling recurring seasonal patterns. These models are particularly relevant to retail series characterized by stable temporal dependence and seasonality.
Prophet: Prophet decomposes a time series into trend, seasonality, and event-related components. It is included because retail demand frequently exhibits recurring weekly or yearly patterns, changing trends, and

calendar-related variations.
Random Forest Regressor: Random Forest represents a nonlinear machine-learning approach. Lagged observations, rolling statistics, and calendar-related variables are transformed into supervised-learning predictors. Its ensemble of decision trees can capture nonlinear relationships and feature interactions without assuming a predefined functional form.
XGBoost/Gradient Boosting: Gradient-boosting models sequentially construct decision trees to correct errors made by preceding learners. XGBoost is included because it can capture complex nonlinear interactions among lagged demand, rolling statistics, promotions, calendar variables, and other explanatory features while incorporating regularization to control model complexity.
LSTM: Long Short-Term Memory is included as the deep-learning candidate because its memory cells and gating mechanisms can capture nonlinear sequential relationships and longer-term temporal dependencies. Its inclusion enables evaluation of whether complex retail series benefit from deep-learning-based forecasting rather than conventional statistical or machine-learning approaches.
All candidate models are trained and evaluated using identical chronological training and validation intervals. RMSE is used as the primary model-selection criterion, while MAE, MAPE, and \(R^2\) are used as complementary forecasting metrics. The candidate obtaining the lowest validation RMSE for a particular retail series is considered its empirically optimal forecasting model.

6.2 Meta-Learner: Random Forest Classifier

The meta-learning component is formulated as a multiclass classification problem, where each class represents one candidate forecasting model. For each retail time series \(D_i\), a meta-feature vector \(\phi(D_i)\) is extracted using characteristics such as trend strength, seasonal strength, autocorrelation, entropy, volatility, distributional statistics, Hurst exponent, and nonlinearity.
These features characterize the underlying behaviour of the series rather than directly predicting future sales. This extends the meta-feature extraction already defined in the proposed framework.Each candidate model \(m\) is independently evaluated on the validation interval, and the model producing the minimum RMSE is assigned as the target meta-label:
$m_i^{*}=\arg\min_\{m\in M\}\ RMSE(m, D_i^{val})$

where \($M =\{\text{ARIMA/SARIMA}$, Prophet, RF, XGBoost, LSTM}\}\). The resulting meta-dataset consists of pairs \((\phi(D_i), m_i^{*})\).
Importantly, the Random Forest Classifier used as the meta-learner is distinct from the Random Forest Regressor in the candidate pool. The regressor predicts future sales, whereas the classifier predicts which forecasting algorithm should be selected. This distinction ensures that the proposed framework represents intelligent algorithm selection rather than stacking and directly addresses the reviewers' concern regarding the inconsistent use of the term “meta-model.”

7. Role of RAG and LLM

The quantitative model-selection mechanism is taken out of the picture, leaving only RAG and the LLM. Meta-learner suggests forecasting algorithm according to the extracted meta-features. The numerical prediction is then obtained using the chosen forecasting algorithm. The LLM uses the retrieved historical model-performance data and domain information to generate a human-readable explanation of the quantitative forecast and evidence from the meta-features, and the LLM uses the retrieved context to generate the explanation.
Therefore, the LLM is not allowed to make an arbitrary choice of the algorithm from the list of predefined candidate-models, but should only use the one chosen by the meta-classifier. This design enables the quantitative evaluation to be deterministic and provides contextual and accessible explanations along with the forecast.
The current prototype includes two different recommendation mechanisms explored based on various experiments: a quantitative Random Forest meta-classifier based on time-series meta-features and a realized RAG–LLM contextual recommender that is limited to Prophet, Random Forest and LSTM. Thus, they are not considered a substitute for the quantitative meta-learner, but they are reported separately.

8. System Implementation



Fig.2. Step by step process for Implementation

8.1. User / Application Layer

This is the starting point of the system. Sales information and forecasting needs are entered by the user, researcher, analyst or retail application. To give an example, the user can request to predict product sales, store demand, weekly revenue, or inventory requirement.

8.2. Automation Layer – n8n

The automation layer is involved in the automation of the workflow. It gathers data, which could be in the form of Excel, databases, APIs, ERP systems, or POS systems. It also initiates preprocessing, model execution, evaluation and report generation automatically.

8.3. AI Processing Layer

This layer examines the retail time series data. It detects missing values, trend, seasonality, noise, outliers, promotional effects, and holiday effects and demand patterns. It also derives meta-features, including data frequency, sales volatility, autocorrelation, seasonal strength, trend strength.

8.4. Vector Knowledge Base- Pinecone.

The knowledge extracted and past forecasting outcomes are stored in a vector database like Pinecone. This layer stores details of the previous datasets, model performance, forecasting rules, business context and research knowledge. With the arrival of a new dataset, the system starts to retrieve similar past cases.

8.5. AI Decision Agent

The brain of the framework is the AI decision agent. It is used to compare the current dataset characteristics and stored knowledge and determines which forecasting model is most appropriate. The examples include selecting Prophet as a strong seasonality, ARIMA/SARIMA as a statistical time series patterns, Random Forest or XGBoost as a multivariate data, and LSTM as complex nonlinear patterns.

8.6. Forecasting Model Layer

The identified model is trained and tested using the data on retail sales. It produces future predictions and performance measurement using performance measures, which include RMSE, MAE, MAPE, R2, and correlation. The system can also compare the various models and pick the best performing model.

8.7. Business Insights Layer

The end product is translated into valuable business information. It offers sales forecasts, demands, inventory

planning suggestions, promotional planning support, and decision-making recommendations. This layer will allow retailers to minimize stockouts, prevent overstocking and enhance business planning.
The figure is an automated meta-forecasting system in intelligent model selection in retail time series analysis. The automation layer is an implementation of n8n which handles the workflow by collecting data, preprocessing it and invoking the forecasting pipeline.
The AI processing layer identifies the data using important statistical and temporal meta-features including trend, seasonality, volatility, outliers and autocorrelation. These features are extracted and then linked with a vector knowledge base like Pinecone which holds records of past forecasting experiments, records of model performance and contextual knowledge.
This retrieved knowledge is used by the AI decision agent to choose the most suitable forecasting model to apply to the given dataset. The chosen forecasting model subsequently makes predictions and measures performance based on conventional forecasting measures.

9. Experimental Setup

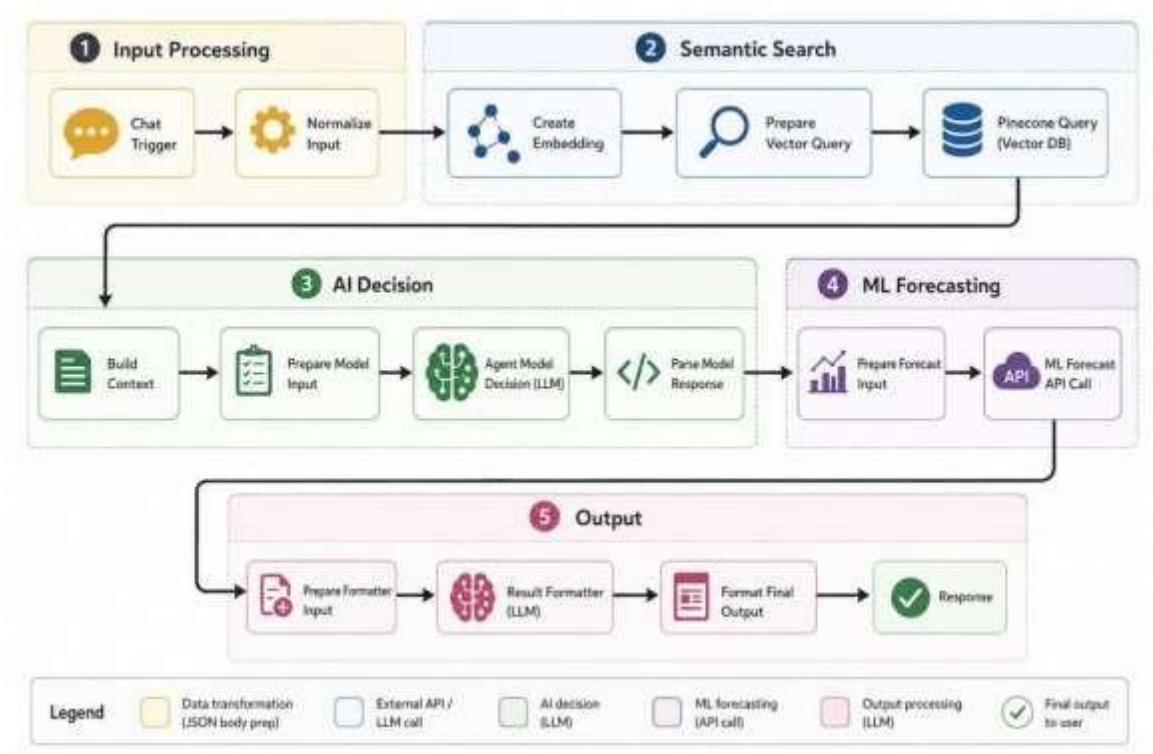


Fig.3. AI-Driven Decision & Forecasting Pipeline

This pipeline is for the hybrid Retrieval-Augmented Generation (RAG) + ML Forecasting Chatbot. Let's now look at each of the 14 steps:

Stages 1 – 2 (Input Processing)

Step 1 — Chat Trigger: The pipeline is triggered by a user's message through a chat application (like a web app or messaging app). This node is waiting for a message and triggers the workflow.

Step 2 — Normalize Input: Perform some cleansing and standardisation, such as removing any special characters or whitespace, and lowercasing the user message — to ensure that the downstream components get a consistent, well-formed text.

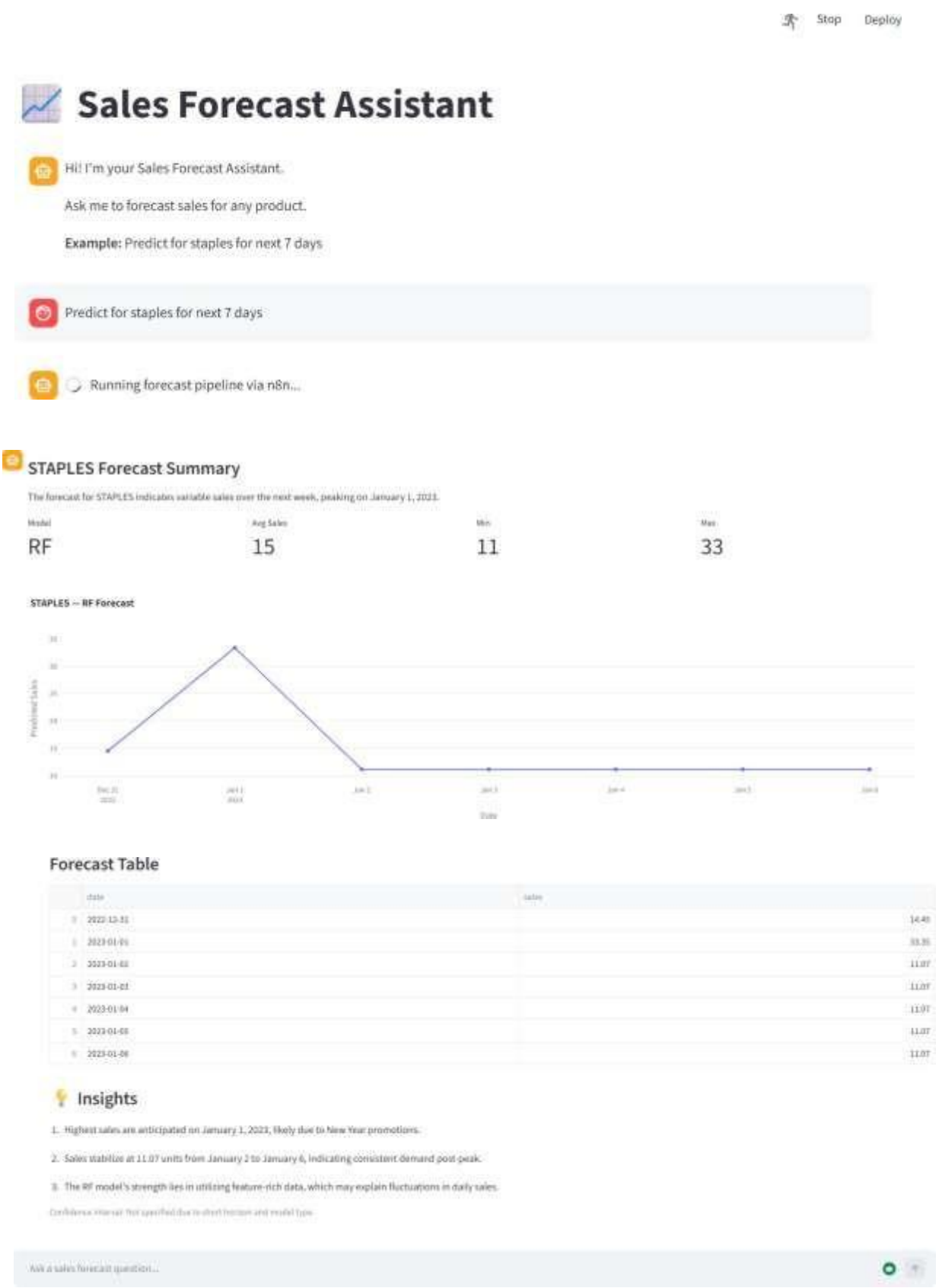
A Semantic Search is performed with the help of a Vector Database (Steps 3–5) in Phase 2.

Step 3 — Create Embedding: The normalized query is passed into an embedding model (OpenAI's API based on the URL in the diagram) that would turn the text into a high dimensional vector representation. This is a numerical representation of the meaning.
Step 4 — Prepare Pinecone Body: A JSON body is created that contains the vector in question as well as any metadata parameters (such as top-k results, namespace, filters) that are needed by the Pinecone vector database API.
Step 5 – Pinecone Query: The vector is sent to a managed vector database, Pinecone. Does an approximate nearest neighbor search and returns the most semantically similar documents or historical records, which may well be historical sales data or domain knowledge.
Teachers incorporate AI-driven decision making in their classrooms (Steps 6-9) (Phase 3).
Step 6 — Build Context: The retrieved Pinecone results are then combined into a structured context block which is essentially the "retrieved knowledge" to be added to the LLM prompt.
Step 7 — Prepare Model Decision Body: A full LLM API request is combined, which consists of:
1. The user's query; 2. Retrieved context; 3. System prompt instructing the model how to interact with the forecasting task.
When assembling the prompt, the following is done: Step 8 — Agent Model Decision: The prompt is sent to an OpenAI LLM (e.g., GPT-4). The model serves as an intelligent agent, and reasons about the context and determines if the answer is simple or requires an ML forecast API call.
Step 9 – Parse Model Response: The raw LLM response (as a JSON or structured text) is transformed into an agent decision, or structured fields or parameters required for the next step.
The next two steps, numbered 10-11, are included in Phase 4 — ML Forecasting.
Step 10 – Prepare Forecast Body: Depending on the decision parsing, a request payload is created for the specific dedicated ML forecasting service (localhost or microservice endpoint: http://api:8000/forec...).
Step 11 — ML Forecast API: The data is sent to a machine learning model specifically designed to predict sales (such as ARIMA, Prophet, or a neural forecasting model) to get quantitative forecasts of sales or projections of trends or signals of anomalies.
The response formatting & output step (Phases 5: Steps 12-14)
Step 12: Create Formatter Body: Encompass the raw ML forecast output in a new prompt payload for a final formatting LLM call with original user query and context.
The second LLM call (also using OpenAI) is Step 13 – Result Formatter (LLM), where the technical forecast numbers are translated into natural, human-readable language, producing an explanation, summary or interpretation of the forecast in human terms.
Step 14 — Format final output: Final post-processing is performed (e.g., markdown, structuring output into the reply for the chat interface), before returning the reply to the user.

10. Result & Discussion

10.1 Results:




Stop
Deploy
Sales Forecast Assistant
Hi! I'm your Sales Forecast Assistant.
Ask me to forecast sales for any product.
Example: Predict for staples for next 7 days
Predict for staples for next 7 days
Running forecast pipeline via n8n...
STAPLES Forecast Summary
The forecast for STAPLES indicates variable sales over the next week, peaking on January 1, 2023.
Model
RF
Avg Sales
15
Min
11
Max
33
STAPLES — RF Forecast
Forecast Table
Insights
1. Highest sales are anticipated on January 1, 2023, likely due to New Year promotions.
2. Sales stabilize at 11.07 units from January 2 to January 6, indicating consistent demand post-peak.
3. The RF model's strength lies in utilizing feature-rich data, which may explain fluctuations in daily sales.

10.2 Forecasting performance across models

Cross Validation Metrics Comparison of machine learning models with traditional statistical models Model – My research work

Model	Method	Correlation	RMSE	MAPE	MAE	R2
Univariate	Facebook Prophet	0.3881	1.2278	0.1610	0.9295	0.148
Multivariate	Random Forest	0.9449	276.61	0.2640	64.87	0.8534
Deep Learning	LSTM	0.9107	247.85	2.9828	122.066	0.8239
Deep Learning	Transformer	- 0.2074	141699.92	12.9526	290.492	-0.16544
Traditional Statistical	ARIMA	- 0.0086	349024.88	10.4822	266.278	- 5.8915

10.3 Uncertainty and Limitations

In the present implementation, main emphasis is placed on the point forecasts. Practical inventory decisions, however, require information about the uncertainty of the forecast in addition to the point-forecast measures of predictive accuracy like RMSE, MAE or MAPE. For future extensions, the probabilistic forecasting or prediction intervals should be included, for example, for tree-based forecasting models Quantile Regression Forests, and for statistical models model-specific interval estimation. Interval coverage and interval width would be better tools for evaluation to assess decision risk.

10.4. Discussion

These experimental results show quantitative evidence of the fact that not all the retail forecasting tasks are homogeneous, and that the suitability of the forecasting models differs. In a set of 68 tasks, 3 different algorithms proved to be the empirically most preferred one, which encouraged the need for automatic model recommendation. The accuracy of the validation-label selection (VLS) for the Random Forest meta-learner was 40.0% and the macro-F1 score was 0.273. For the test intervals that were not held out, it equaled the oracle model on 50.0% of the tasks, and had a mean RMSE of 259.349.

Compared to random selection, the mean RMSE of the forecasts and mean selection regret were reduced by a significant margin with the meta-learning approach (from 316.931 to 259.349 for the former, from 68.997 to 11.415 for the latter). A lower RMSE of 250.967 and a regret of 3.032 were obtained by the global-best SGD baseline. The present experiment therefore doesn't show the superiority of the proposed meta-learner in general. On the contrary, it shows that the process of model suitability is heterogeneous and partially learnable from the characteristics of the time-series, while the process of cross-series generalization is still difficult.

The feature importance analysis also shows that sparsity, coefficient of variation, mean sales, skewness, weekly autocorrelation and trend characteristics are important features that are included in the model recommendation. However, independent retail sub-categories were few in number, and there was a class imbalance in optimal-model labels, as well as the possibility of distribution differences between the sub-categories which constrain generalizability. Much larger cross-dataset experiments would thus be needed before determining robust universal model-selection performance.

The manuscript has been extensively restructured and edited in terms of language. Evaluation of forecasting models is now reported separately, as is the implementation of RAG–LLM with meta-learning evaluation. All informal language has been eliminated, mathematical notation has been standardized, and the Discussion and Conclusion now differentiate between what has shown to be true in experiments and what is intended for future.

Table.5. Meta-feature importance

Meta-feature	Importance
Zero-sales ratio	0.0908
Coefficient of variation	0.0817
Mean	0.0786
Skewness	0.0764
ACF lag 7	0.0747
Trend strength	0.0727
Kurtosis	0.0723
ACF lag 14	0.0722

For better interpretability of the meta-learning mechanism, feature-importance analysis was included. Sparsity, variability, distributional features, weekly autocorrelation and trend information were the most influential features. The current meta-learner was also found not to outperform the global-best baseline in the Discussion, indicating that more independent time series and more comprehensive meta-features as well as more cross-dataset training are needed to enhance its generalization.

11. Conclusion

This study has proposed and tested an intelligent retail meta-forecasting perspective that includes: Forecasting-model evaluation, Meta-learning-based recommendation, and a deployed forecasting and recommendation pipeline based on RAG-LLM. 68 forecasting tasks were created from 17 retail sub-category series in the experimental analysis and it was found that none of the candidates forecasting algorithms performed optimally on all of the tasks. Ridge, SGD and Random Forest were optimal for 29, 26 and 13 tasks, respectively.

The Random Forest meta-learner achieved 0.400 accuracy in selecting the retail series and a macro-F1 of 0.273 on unseen retail series. The RMSE and regret values were 259.349 and 11.415, respectively, significantly outperforming random selection, but were still much worse than the global-best SGD baseline, which had an RMSE of 250.967 and a regret of 3.032. Hence, the findings validate the motivation of intelligent model recommendation and also show that the cross-series generalization is still a tough task.

The prototype developed for the RAG-LLM is a qualitative extension of the experiments mentioned above, as it provides semantic retrieval as well as constrained contextual model recommendation and natural language interpretation. Moving forward, it is essential to test the impact of extending the meta-training data to independent retail data, investigate a wider range of forecasting-model pools, explore the use of probabilistic forecasting and uncertainty quantification, and systematically compare meta-learning and contextual LLM recommendation mechanisms.

Bibliography

1. Taylor, S.J.; and Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37–45.
2. de Castro Moraes, T.; Yuan, X.M.; and Chew, E.P. (2024). Hybrid convolutional long short-term memory models for sales forecasting in retail. Journal of Forecasting, 43(5), 1278–1293.
3. Ma, S.; and Fildes, R. (2021). Retail sales forecasting with meta-learning. European Journal of Operational Research, 288(1), 111–128.
4. Talkhi, N.; Akhavan Fatemi, N.; Jabbari Nooghabi, M.; Soltani, E.; and Jabbari Nooghabi, A. (2024). Using meta-learning to recommend an appropriate time-series forecasting model. BMC Public Health, 24, 148.
5. Chen, S.; Ke, S.; Han, S.; Gupta, S.; and Sivarajah, U. (2024). Which product description phrases affect sales forecasting? An explainable AI framework by integrating WaveNet neural network models with multiple regression. Decision Support Systems, 176, 114065.
6. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.
7. Tyralis, H.; and Papacharalampous, G.A. (2017). Variable selection in time series forecasting using random forests. Algorithms, 10(4), 114.
8. Seaman, B. (2018). Considerations of a retail forecasting practitioner. International Journal of Forecasting, 34(4), 822–829.
9. Meinshausen, N. (2006). Quantile regression forests. Journal of Machine Learning Research, 7, 983–999.