Benchmarking Tabular Foundation Models for Agricultural Yield Prediction

Mohammed Musthafa Rafi[1], **Timilehin T. Ayanlade**[1], **Baskar Ganapathysubramanian**[1], **Soumik Sarkar**[1], **Adarsh Krishnamurthy**[1], **Chinmay Hegde**[2], **Aditya Balu**[1]

[1]Iowa State University, [2]New York University
{mohd7, ayanlade, baskarg, soumiks, adarsh}@iastate.edu,
chinmay.h@nyu.edu, baditya@iastate.edu

Abstract

Accurate crop yield prediction is crucial for global food security and agricultural planning. This study benchmarks modern tabular foundation models and automated machine learning frameworks across three diverse agricultural datasets: (1) soybean yields with 86,101 temporal sequences, (2) global multi-crop data with 28,242 samples across 101 countries, and (3) EU-27 regional crops with 8,656 samples and significant missing data. We evaluate TabPFNv2 (an improved implementation of the TabPFN architecture), AutoGluon, and PyCaret to determine which approach works best under different data conditions. Our results show that model performance is highly context-dependent. AutoGluon performs best on large-scale complete data, PyCaret performs well on diverse multi-crop scenarios, while TabPFNv2 demonstrates distinct advantages on datasets with missing values (about a two percentage point gain in R^2 on EU-27). These findings show that none of the tested methods are universally superior. Furthermore, foundation models provide robust zero-shot predictions, particularly while handling incomplete data, which is essential for practical agricultural AI deployment.

Introduction

Predicting crop yields accurately is essential for food security and agricultural planning, yet remains challenging due to complex climate-yield interactions, temporal weather patterns, and inconsistent data quality across regions.

Machine learning approaches for crop yield prediction have evolved from simple regression models to sophisticated deep learning architectures (Chlingaryan, Sukkarieh, and Whelan 2018; van Klompenburg, Kassahun, and Catal 2023). Early work demonstrated that CNN–RNN models could achieve $R^2 \approx 0.75$ by capturing temporal patterns in weather data (Khaki and Wang 2019). Other studies integrated satellite imagery with machine learning for wheat and soybean forecasting, showing the value of multi-modal data integration (Cai et al. 2019; Schwalbert et al. 2020; You et al. 2017; Nevavuori, Narra, and Lipping 2019; Fan et al. 2022). Ensemble methods have been shown to consistently outperform single models by 10–20% (Shahhosseini et al. 2021), but most of these works still focus on a single crop or region and require extensive labeled data.

Foundation models, which are large-scale models pretrained on broad distributions and adapted with little or no fine-tuning, have recently emerged as a promising alternative. While they have revolutionized natural language processing and computer vision, their extension to structured tabular data is more challenging: features differ in type and scale, and do not share the spatial or sequential structure exploited in images or text.

Within this emerging space, TabPFNv2 is a representative tabular foundation model, extending the original TabPFN architecture. TabPFNv2 is the officially improved implementation of the TabPFN architecture, as stated by the original authors in the v2 release notes (Hollmann et al. 2023). TabPFNv2 inherits the in-context learning formulation while improving robustness and efficiency when operating at its full 10,000-row context window on heterogeneous tabular datasets.

In this work, we provide the first comprehensive benchmark comparing tabular foundation models against AutoML frameworks for agricultural yield prediction across three diverse datasets. Our main contributions are:

1. **Systematic evaluation of tabular foundation models for agriculture:** We conduct a multi-dataset comparison of TabPFNv2 against state-of-the-art AutoML frameworks (AutoGluon, PyCaret) across diverse agricultural contexts, revealing when foundation models outperform traditional approaches.
2. **Comprehensive preprocessing pipelines tailored to agricultural data:** We develop and validate dataset-specific preprocessing strategies that handle temporal aggregation, missing values, and feature engineering, providing reproducible baselines for future research.
3. **Empirical insights on model selection under data constraints:** Through quantitative analysis across 86,101 soybean sequences, 28,242 global samples, and 8,656 EU regional records, we observe that TabPFNv2 performs well with missing data (2.18% improvement), while AutoML performs best on large-scale complete datasets, providing practical guidelines for model selection.

Accepted at the First International Workshop on AI in Agriculture (Agri AI), co-located with AAAI 2026.

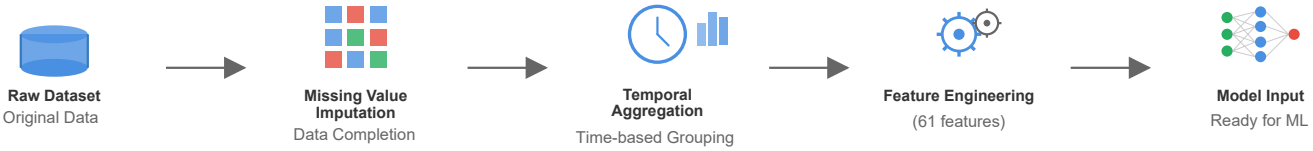

Figure 1: Overview of the benchmarking pipeline: Three diverse agricultural datasets are processed through tailored preprocessing strategies and evaluated using tabular foundation models (TabPFNv2) and AutoML frameworks (AutoGluon, PyCaret) to assess performance under varying data conditions.

Methodology

Our methodology evaluates the effectiveness of tabular foundation models versus automated machine learning frameworks for agricultural yield prediction. The core hypothesis is that foundation models, through their pre-trained representations, can better generalize to agricultural data with limited samples or missing values, whereas AutoML frameworks excel when sufficient complete data enables thorough algorithm exploration and ensemble construction.

To test this hypothesis, we design a comprehensive benchmarking pipeline that: (1) processes three diverse agricultural datasets representing different scales, crop types, and data quality scenarios; (2) applies tailored preprocessing strategies to handle domain-specific challenges such as temporal aggregation and missing values; (3) evaluates both foundation-model and AutoML approaches under consistent conditions; and (4) analyzes performance patterns to identify when each approach is most effective. Figure 1 provides an overview of the full pipeline.

Concretely, we ask: (i) when do tabular foundation models match or surpass AutoML frameworks, and (ii) how do dataset size, completeness, and feature composition influence this trade-off?

Tabular Foundation Models

Traditional machine learning on tabular data requires training a new model from scratch for each task, demanding substantial labeled data and computational resources. Foundation models fundamentally change this paradigm by leveraging pre-training on vast, diverse datasets to learn generalizable representations that transfer to new tasks with minimal adaptation.

For tabular data, this presents unique challenges compared to text or images. Tabular features are heterogeneous—mixing numerical, categorical, and ordinal types with different scales and distributions. Features lack the spatial or sequential structure that convolutional or recurrent networks naturally exploit. Moreover, the semantic meaning of features varies dramatically across domains; "temperature" in weather data has no relation to "temperature" in industrial processes.

TabPFNv2 (part of the TabPFN family (Hollmann et al. 2023)) addresses these challenges through an innovative approach: instead of pre-training on real tabular datasets, it trains on millions of synthetically generated datasets drawn from a carefully designed prior distribution. This synthetic pre-training strategy offers several advantages:

Architectural Design TabPFNv2 employs a transformer encoder that processes both features and labels jointly. For a dataset with n samples and d features, the input is formatted as a sequence where each position contains a feature vector and its corresponding label. The transformer learns to predict labels for unlabeled samples based on the patterns observed in labeled samples within the same context.

Synthetic Pre-training During pre-training, TabPFNv2 is exposed to datasets generated from a prior that combines Bayesian neural networks with varying architectures, diverse feature distributions (Gaussian, categorical, and mixed types), a wide range of dataset sizes and feature dimensions, different noise levels and missing-data patterns, and both classification and regression targets. This diversity ensures that the model encounters a broad spectrum of tabular patterns and learns meta-features that generalize across domains.

In-Context Learning At inference, TabPFNv2 performs in-context learning—it receives the entire training set (up to 10,000 samples per context window) along with test samples in a single forward pass. The model implicitly performs the equivalent of training and prediction simultaneously, without any gradient updates. This is formalized as:

$$P(y_{\text{test}}|X_{\text{test}}, D_{\text{train}}) = \text{TabPFNv2}([X_{\text{train}}, y_{\text{train}}, X_{\text{test}}])$$

where $D_{\text{train}} = \{X_{\text{train}}, y_{\text{train}}\}$ is the training data provided as context. This process is illustrated in Figure 2 for the agricultural yield prediction context.

These design choices yield several benefits in agricultural settings. TabPFNv2 can be deployed in a nearly zero-shot fashion, without task-specific fine-tuning, which is useful for new crops or regions with limited historical data. Its pre-training setup makes it sample-efficient and robust to missing values, both of which are common in sensor-based field experiments. In addition, the model exposes prediction uncertainties, which are important for risk-aware decision making in agricultural planning.

AutoML Frameworks

In contrast to foundation models, AutoML frameworks take an exhaustive search approach, systematically exploring algorithm spaces and hyperparameter configurations:

AutoGluon constructs a multi-layer stacked ensemble in which Level-1 models (including Random Forest (Breiman 2001), Extra Trees, LightGBM (Ke et al. 2017), CatBoost (Prokhorenkova et al. 2018), XGBoost (Chen and

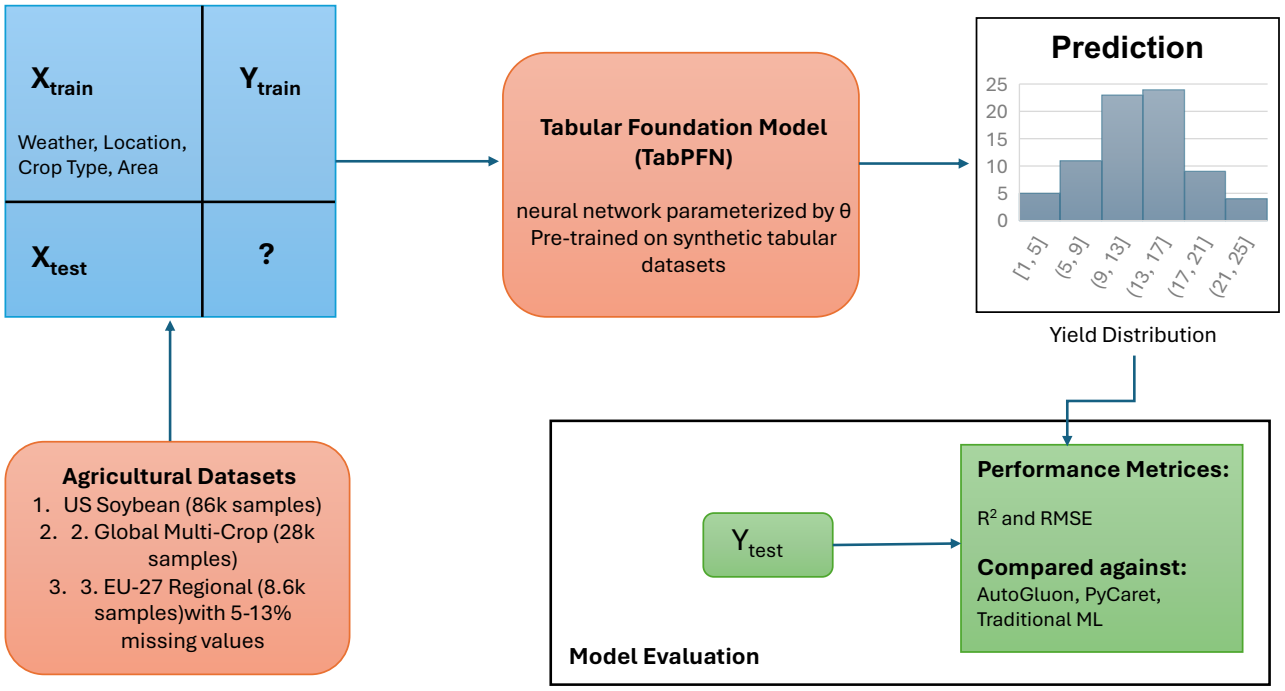

Figure 2: Schematic of TabPFNv2's in-context learning procedure applied to agricultural yield prediction. The model receives the full training set and test samples in a single context and outputs predictions without task-specific gradient updates.

Guestrin 2016), and simple neural networks) are trained independently, and their predictions are then combined by Level-2 meta-learners. The framework automatically performs feature preprocessing, handles missing values, and selects the final ensemble based on validation performance.

PyCaret exposes a unified interface for tabular regression, wrapping a collection of standard models under a common preprocessing and evaluation pipeline. In our experiments, it evaluates candidate regressors using cross-validation under a fixed preprocessing recipe (normalization, encoding, and basic feature engineering) and selects the best-performing model according to the specified metric.

Experimental Protocol

All experiments follow a consistent evaluation protocol to ensure fair comparison across models and datasets. For the soybean dataset, which contains temporal sequences spanning multiple growing seasons, we adopt a chronological split with 80% of samples for training, 10% for validation, and 10% for testing, ordered by time to prevent data leakage. For the global and EU-27 datasets, which lack inherent temporal ordering, we employ stratified 5-fold cross-validation to ensure representative distribution of crops and regions across splits, and report mean performance metrics across all folds.

Numerical features are standardized using z-score normalization with mean zero and unit variance, where scaling parameters are computed solely on the training partition to avoid information leakage. Missing values are handled in a model-aware fashion: for TabPFNv2 and gradient-boosting methods with native missing value support (XGBoost, CatBoost, HistGradientBoosting), missing entries are passed directly to the model, allowing the algorithm to learn from missingness patterns. For models requiring complete data (Random Forest, Linear Regression), we apply mean imputation using statistics computed on the training set only. We do not apply any transformation to the target variable to maintain direct interpretability of yield predictions in their original units.

TabPFNv2 operates in zero-shot mode without any fine-tuning or hyperparameter search. For datasets exceeding the 10,000-sample context window (soybean and global), we apply stratified sampling that preserves the marginal distribution of the target variable, ensuring that the subsampled training set remains representative. The model uses its pre-trained 32-ensemble configuration throughout. For the discretized variant, continuous yield values are binned into five equal-frequency bins using scikit-learn's `KBinsDiscretizer` with quantile strategy, converting the regression task into multi-class classification compatible with TabPFNv2's original formulation.

All experiments were conducted on an Alienware Aurora R16 workstation equipped with an Intel Core i9-14900KF processor (24 cores, 32 threads, 3.20 GHz), 64 GB RAM, and an NVIDIA GeForce RTX 4070 SUPER GPU. TabPFNv2 inference completed in under one second per dataset across all experiments. AutoGluon training time varied from approximately 5 minutes on EU-27 with medium-quality preset to 60 minutes on the soybean dataset with best-quality preset. PyCaret typically completed within 20-30 seconds due to its simpler model comparison pipeline.

Table 1: Summary of dataset characteristics and associated preprocessing requirements.

Block	Characteristic	Dataset		
		Soybean	**Global**	**EU-27**
Volume	Total samples	86,101	28,242	8,656
	Features (after preprocessing)	61	6	8
Data properties	Temporal aggregation applied	Yes	No	No
	Missing values (%)	8.3	0.0	5–13
	Crops covered	1	10	20
Coverage & target	Geographical scope	US Midwest	101 countries	27 EU countries
	Target unit	bu/acre	hg/ha	tonnes/ha

Datasets and Preprocessing

We evaluate our methods on three publicly available datasets that capture complementary agricultural scenarios: large-scale single-crop prediction in the US, multi-crop global production, and regional EU production with incomplete records. Table 1 summarizes the key characteristics of each dataset and their preprocessing requirements.

The first dataset, *US Soybean Yield*, contains 86,101 samples with daily weather sequences (60–180 days) linked to end-of-season soybean yields. Each record combines temporal weather observations with geographic and varietal information, so that daily temperature and precipitation must be aggregated into fixed-length tabular features.

The second dataset, *Global Multi-Crop (FAO)*, includes 28,242 samples spanning ten major crops across 101 countries from 1990 to 2013. The features consist of aggregate climatic variables (rainfall, temperature), management indicators such as pesticide usage, and categorical encodings for crop and country. This dataset is fully observed and does not contain missing values.

The third dataset, *EU-27 Regional Crops*, comprises 8,656 samples for twenty crops across 27 EU countries at the NUTS regional level. In contrast to the global dataset, it exhibits 5–13% missing values across features, reflecting typical gaps in regional reporting and data collection.

For the soybean dataset, we convert daily weather time series into fixed-length tabular descriptors by computing simple statistical summaries (mean, standard deviation, and related moments) over the growing season for each weather variable. This yields 61 engineered features that capture both average conditions and intra-season variability while keeping the representation compatible with tabular models.

Missing values are handled in a model-aware manner. For TabPFNv2 and gradient-boosting methods with native support for missing entries, we pass missing values directly to the model, allowing it to exploit missingness patterns as part of the learning process. For algorithms that do not accept missing values (e.g., Random Forest and Linear Regression), we perform mean imputation using statistics computed on the training split only, to avoid leakage into the validation and test sets.

Finally, we account for TabPFNv2's 10,000-sample context window. For datasets exceeding this size (soybean and global), we draw a stratified subsample of the training data that preserves the marginal yield distribution, and use the full held-out test set for evaluation. The EU-27 dataset is small enough to be used in full without subsampling.

Model Configuration and Training

For TabPFNv2, we use the publicly released 32-ensemble model with default hyperparameters and no task-specific fine-tuning. On datasets with more than 10,000 training samples, we apply the stratified sampling procedure described above. Because models in the TabPFN family were originally developed for classification, we also experiment with a discretized version of the soybean task in which the continuous yield is discretized into five bins using `KBinsDiscretizer` with a quantile strategy, converting the regression problem into a classification task for TabPFNv2.

For AutoGluon, we adopt different presets depending on dataset size. On the large soybean dataset, we use `preset="best_quality"` to enable deeper ensembling and more extensive hyperparameter search. On the global and EU-27 datasets, we use `preset="medium_quality"` to reduce training time while retaining stacking across multiple base learners. Time limits are chosen between 300 and 3600 seconds, scaled by dataset size, with 5–10 bagging folds for variance reduction.

For PyCaret, we use the standard regression workflow with 5-fold cross-validation. All available regression models are compared under the default preprocessing pipeline (normalization, encoding, and basic feature engineering), and the best-performing model is selected using RMSE as the optimization metric (`optimize="RMSE"`).

To contextualize the performance of TabPFNv2 and the AutoML frameworks, we also report results for a set of commonly used baselines: Random Forest, Gradient Boosting, Histogram Gradient Boosting, XGBoost, and Linear Regression, each with standard library defaults and minimal tuning. Gradient Boosting refers to `GradientBoostingRegressor`, while Histogram Gradient Boosting refers to `HistGradientBoostingRegressor`, both from scikit-learn. These baselines serve both as sanity checks and as reference points for the observed gains from foundation models and AutoML.

Results

We compare the performance of TabPFNv2 and the two AutoML frameworks on the three agricultural datasets. We report R^2 and RMSE to interpret the patterns in the context of each dataset's scale, feature composition, and data quality.

US Soybean Dataset

The soybean dataset represents a large-scale setting with complete temporal weather records and 86,101 samples. Table 2 summarizes the model performance. AutoGluon achieves the best overall accuracy with $R^2 = 0.8462$ and RMSE of 5.84 bu/acre, outperforming all other models. The gap of 8.8% in R^2 between AutoGluon and TabPFNv2 suggests that with large, complete datasets, AutoML ensembles may better exploit available training data.

The feature importance analysis from the AutoGluon ensemble indicates that yield is driven more by variability than by mean conditions. The standard deviation of temperature over the growing season is assigned the highest importance (0.31), followed by precipitation-related features (0.24), latitude (0.18), and maturity group (0.15). This is consistent with prior agronomic findings on the role of weather variability during critical growth stages.

TabPFNv2 shows reasonable performance, though the 10,000-sample context window limits its ability to use the full dataset. Using stratified sampling, the model attains strong performance in the regression setting, with marginal improvement when the target is discretized into five yield bins. The small improvement under quantization is consistent with the TabPFN family being originally designed for classification tasks. Nevertheless, the sampling constraint prevents TabPFNv2 from fully exploiting the richness of the 86k-sample dataset.

Traditional baselines provide additional context. XGBoost and Random Forest show competitive performance, while Gradient Boosting and Linear Regression trail further behind. These results suggest that stacked ensembles can leverage complementary base learners more effectively than individual models on this dataset.

Global Multi-Crop Dataset

The global multi-crop dataset spans 10 crops and 101 countries, where categorical features (crop type, country) may provide informative signals. Table 3 shows that all advanced methods achieve high accuracy, with R^2 exceeding 0.97 in most cases.

PyCaret attains the best performance with $R^2 = 0.9869$ and RMSE of 9,762.65 hg/ha, slightly ahead of AutoGluon.

TabPFNv2 achieves comparable accuracy using only 35% of the training data due to its context window limit. This suggests potential sample efficiency, though the performance gap indicates room for improvement.

From a computational efficiency perspective, PyCaret completes its full pipeline in 21.76 s, while AutoGluon requires 118.31 s for its more extensive ensemble construction. TabPFNv2 achieves inference in just 0.39 s, demonstrating its advantage for rapid deployment scenarios.

Feature importance analysis for the tree-based models shows that crop type is the dominant predictor (importance around 0.42), followed by temperature (0.21), pesticide usage (0.18), and rainfall (0.13). This suggests that, at a globally aggregated level, intrinsic crop characteristics and management practices overshadow fine-grained environmental variations.

EU-27 Regional Crops Dataset

The EU-27 dataset provides a contrasting scenario with moderate size (8,656 samples), higher crop diversity (20 crops), and 5–13% missing values. This makes it a natural test bed for TabPFNv2's robustness to incomplete tabular data. The results are summarized in Table 4.

On this dataset, TabPFNv2 achieves the highest accuracy with $R^2 = 0.9107$ and RMSE of 12.44 tonnes/ha, outperforming both AutoGluon and PyCaret's top model (CatBoost). The 2.18% improvement in R^2 over the next best method may reflect benefits from pre-training on datasets with missing value patterns. Because the full dataset fits

Table 2: Model performance on the soybean dataset (86,101 samples).

Model	R^2	RMSE (bu/acre)
AutoGluon	**0.8462**	**5.84**
TabPFNv2 (discretized)	0.7784	7.01
TabPFNv2 (baseline)	0.7759	7.05
Random Forest	0.7012	8.14
XGBoost	0.7234	7.83
Gradient Boosting	0.6893	8.31
Linear Regression	0.5876	9.56

Table 3: Model performance on the global multi-crop dataset.

Model	R^2	RMSE (hg/ha)
PyCaret (ExtraTrees)	**0.9869**	**9,762.65**
AutoGluon	0.9835	10,972.29
Random Forest	0.9794	12,240.56
TabPFNv2 (discretized)	0.9716	14,163.17
Gradient Boosting	0.9387	21,137.47
XGBoost	0.9521	18,693.25
Linear Regression	0.0856	81,597.44

Table 4: Model performance on the EU-27 regional crops dataset.

Model	R^2	RMSE (tonnes/ha)
TabPFNv2 (discretized)	**0.9107**	**12.44**
PyCaret (CatBoost)	0.8914	13.72
Histogram Gradient Boosting	0.8913	13.73
AutoGluon	0.8893	13.85
XGBoost	0.8519	16.02
Random Forest	0.8360	16.86
Linear Regression	0.3124	34.52

within the 10k-sample context window, TabPFNv2 can use all samples simultaneously at inference time without subsampling.

The comparison with tree-based baselines sheds light on the role of missing-value handling strategies. Models with native support for missing values—including TabPFNv2, HistGradient Boosting, XGBoost, and CatBoost—demonstrate consistently strong performance, with TabPFNv2 leading. In contrast, Random Forest and Linear Regression, which rely on mean imputation, show notably degraded performance. This pattern suggests that native handling of missing values may be beneficial in this setting. TabPFNv2's pre-training on synthetic datasets with diverse missing-data patterns could contribute to its performance here.

The EU-27 results suggest conditions under which TabPFNv2 may be effective: moderate dataset size that fits within the context window, and missing values that challenge imputation-based approaches.

Cross-Dataset Trends

The cross-dataset trends in Tables 2, 3, and 4 reveal several key patterns. On the large, fully observed soybean dataset, AutoGluon performs best, consistent with advantages of deep ensembling when sufficient samples are available. On the global multi-crop dataset, both AutoML frameworks and TabPFNv2 perform well, suggesting that multiple approaches can perform well when informative categorical features are present.

The EU-27 dataset highlights a complementary regime where TabPFNv2 is particularly effective. Here, the combination of moderate dataset size and non-trivial missingness favors a model that has meta-learned how to handle incomplete and heterogeneous tabular inputs. Across all three datasets, TabPFNv2 offers the fastest training and inference times (on the order of seconds), while AutoGluon provides the highest accuracy on the largest and cleanest dataset at the cost of significantly higher computation, underscoring that no single approach is universally best.

Discussion and Conclusions

Our results suggest that performance depends on dataset size, completeness, and feature composition—addressing the questions posed in the Methodology section.

On the large, fully observed soybean dataset, AutoGluon achieves the best performance, improving R^2 by 8.8% over TabPFNv2. This is consistent with prior observations that ensembles can be effective on large tabular datasets with complete data. When many complete samples are available and training time is not a bottleneck, high-quality AutoML pipelines are a natural choice.

In contrast, on the EU-27 regional dataset with 5–13% missing values and moderate sample size, TabPFNv2 outperforms both AutoGluon and PyCaret, improving R^2 by 2.18%. This suggests conditions where tabular foundation models may offer advantages: moderate size within the context window and incomplete records. The global multi-crop dataset lies between these extremes. Here, all advanced models reach high performance, with PyCaret slightly ahead, suggesting that informative categorical features (crop, country) may make the task easier for a range of methods.

From a practical standpoint, these patterns suggest tentative guidelines. For large, complete datasets where training time is not a constraint, AutoML frameworks may be preferable. When data are limited or contain missing values, tabular foundation models may provide reasonable performance with minimal tuning. In intermediate regimes, both approaches are viable, and considerations such as deployment constraints, interpretability, and available compute become more important. Hybrid strategies are also promising, for example using TabPFNv2 for rapid screening of feature sets or regions, followed by AutoML ensembles for final model selection.

A practical advantage of TabPFNv2 is its computational footprint. Across all three datasets, TabPFNv2 completes inference in under one second, while AutoGluon requires 5 to 60 minutes of training depending on dataset size and ensemble configuration. This speedup may be useful for rapid prototyping or scenarios requiring frequent updates. However, the 10,000-sample context window necessitates stratified subsampling on larger datasets and prevents TabPFNv2 from fully exploiting very large training sets.

This study has several limitations. We focus on a single foundation model (TabPFNv2) and two AutoML frameworks; other recent tabular architectures may exhibit different behavior under the same protocol. The current TabPFNv2 implementation also enforces a 10,000-sample context limit, which prevents straightforward use on very large datasets without subsampling. Finally, our temporal aggregation deliberately simplifies daily weather into summary statistics, which discards finer-grained subseasonal patterns. Extending this work to domain-specific pre-training, larger-context tabular foundation models, and time-series foundation models such as Chronos (Ansari et al. 2024) and AutoGluon-TimeSeries (Shchur et al. 2023) is a natural next step.

In summary, we provide a systematic comparison of tabular foundation models and AutoML frameworks for agricultural yield prediction across three representative datasets. We observe that AutoGluon and PyCaret perform well on large, complete datasets, while TabPFNv2 shows advantages on moderate-sized datasets with missing values. Both approaches outperform standard baselines in our experiments. We release our preprocessing pipelines, experimental scripts, and processed datasets at https://github.com/itsMustafamr/Yield.git to support reproducibility and to facilitate future work at the intersection of foundation models and agricultural AI.

Acknowledgments

This work is supported by the AI Research Institutes program [*AI Institute for Resilient Agriculture (AIIRA)*, Award No. 2021-67021-35329] from the National Science Foundation and U.S. Department of Agriculture's National Institute of Food and Agriculture. We also acknowledge the support from the Plant Science Institute.

References

Ansari, A. F.; Stella, L.; Türkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Pineda Arango, S.; Kapoor, S.; Zschiegner, J.; Maddix, D. C.; Wang, H.; Mahoney, M. W.; Torkkola, K.; Wilson, A. G.; Bohlke-Schneider, M.; and Wang, Y. 2024. Chronos: Learning the Language of Time Series. *arXiv preprint arXiv:2403.07815*.

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32.

Cai, Y.; Guan, K.; Lobell, D.; Potgieter, A. B.; Wang, S.; Peng, J.; Xu, T.; Asseng, S.; Zhang, Y.; You, L.; and Peng, B. 2019. Integrating satellite and climate data to predict wheat yield in Australia using machine learning approaches. *Agricultural and Forest Meteorology*, 274: 144–159.

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chlingaryan, A.; Sukkarieh, S.; and Whelan, B. 2018. Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review. *Computers and Electronics in Agriculture*, 151: 61–69.

Fan, J.; Bai, J.; Li, Z.; Ortiz-Bobea, A.; and Gomes, C. P. 2022. Graph Neural Networks for Spatial-Temporal Crop Yield Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3679–3687.

Hollmann, N.; Müller, S.; Eggensperger, K.; and Hutter, F. 2023. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *Proceedings of the 11th International Conference on Learning Representations*.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, volume 30.

Khaki, S.; and Wang, L. 2019. Crop Yield Prediction Using Deep Neural Networks. *Frontiers in Plant Science*, 10: 621.

Nevavuori, P.; Narra, N.; and Lipping, T. 2019. Crop yield prediction with deep convolutional neural networks. *Computers and Electronics in Agriculture*, 163: 104859.

Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; and Gulin, A. 2018. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31.

Schwalbert, R. A.; Amado, T.; Corassa, G.; Pott, L. P.; Prasad, P. V.; and Ciampitti, I. A. 2020. Satellite-based soybean yield forecast: Integrating machine learning and weather data for improving crop yield prediction in southern Brazil. *Agricultural and Forest Meteorology*, 284: 107886.

Shahhosseini, M.; Hu, G.; Huber, I.; and Archontoulis, S. V. 2021. Forecasting Corn Yield With Machine Learning Ensembles. *Frontiers in Plant Science*, 12: 709008.

Shchur, O.; Turkmen, C.; Erickson, N.; Shen, H.; Shirkov, A.; Hu, T.; and Wang, Y. 2023. AutoGluon-TimeSeries: AutoML for Probabilistic Time Series Forecasting. *arXiv preprint arXiv:2308.05566*.

van Klompenburg, T.; Kassahun, A.; and Catal, C. 2023. Machine learning for crop yield prediction: A systematic literature review. *Computers and Electronics in Agriculture*, 177: 105709.

You, J.; Li, X.; Low, M.; Lobell, D.; and Ermon, S. 2017. Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.