AI-Driven Stock Recommendation: Optimizing Predictions with Hyperparameter Tuning

Anvith S G
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4cse22007@bl.students.amrita.edu

Nithish Kushal Reddy
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4cse22043@bl.students.amrita.edu

Sreelakshmi C S
Dept. of Computer Science Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
cs_sreelakshmi@blr.amrita.edu

***Abstract*—Transforms large datasets into actionable insight that helps investors make the most out of it. Therefore, this study aims at the development of an AI-based stock recommendation system using the principle of value investing and sorting the stocks as ”Not Recommended,” ”Recommended,” and ”Highly Recommended.” The dataset is a detailed dataset from Ticker Tape comprising financial ratios, market trends, and sectoral benchmarks, which has been preprocessed using Winsorization, imputation, and feature scaling for machine learning. The metrics used were accuracy, F1-score, and AUC-ROC for the nine machine learning models evaluated: Random Forest, SVC, Decision Tree, AdaBoost, XGBoost, Bagging, Voting, Stacking, and KNN. Hyperparameter tuning with Optuna has improved model performance. XGBoost with the highest accuracy at 90.48% and AUC-ROC at 97.45% were obtained. The system is built as a Flask-based web application and utilizes the XGBoost model and preprocessing pipeline for real-time prediction. This project shows how AI agents, combining advanced models with optimization techniques, can assist an investor in making value investing decisions.**

***Index Terms*—Artificial Intelligence, Stock Recommendation, Machine Learning, Value Investing, XGBoost, Hyperparameter Tuning, Optuna, Ensemble Methods, Flask Deployment, Financial Data Analysis**

I. Introduction

Artificial Intelligence (AI) is revolutionizing industries by giving systems the ability to mimic the cognitive functions associated with human intelligence, like problem-solving, decision-making, and learning. In the financial realm, AI has emerged as a powerful tool for stock analysis and recommendation, using huge datasets and computational power to provide insights that were otherwise unattainable. This paper explores the application of AI in the development of a stock recommendation system based on the value investing principles, in which long-term investments go into fundamentally strong companies. The AI system basically revolves around an agent-based architecture. This project builds an AI agent that operates as a clever decision-making entity processing financial data and determining the best recommendations for the stocks. This agent performs in a highly complex environment, involving moving market conditions, historical stock performances, and the fundamental indicators including return on equity (ROE), EBITDA margin, and debt-to-equity ratio. These variables are percepts observable elements of the environment that an agent captures to analyze and reason about the stock market. The objective of this AI system is to classify stocks into three categories: (1) not recommended, (2) recommended, and (3) highly recommended. This categorization is aligned with value investing principles that help identify undervalued stocks with a strong growth prospect. The journey of the AI starts from the start state represented by the raw dataset acquired from Ticker Tape containing financial metrics, market trends, and sector-specific insights. The agent makes the data ready for analysis by using the preprocessing technique of Winsorization and imputing missing values. This the AI does by its actuators-machine learning models implemented in making their predictions. These are just a few models in that class: Random Forest, Support Vector Machines, Decision Trees, AdaBoost, XGBoost, etc. By the application of deep methods in hyperparameter optimization such as Optuna, these models allow AI the ability to sharpen decision making powers as it at the same time enhances predictivity. This paper deals with the design, development, and evaluation of an AI-based stock recommendation system. For this purpose, it also refers to some key concepts of AI-agents, percepts, goal states, and actuators-and preprocessing and tuning mechanisms. This system, which is based on value investing principles, shows how AI bridges the gap between traditional investment strategies and modern computational techniques by providing a strong solution for stock selection in the Indian market.

II. Litureature Survey

Development of a value investment AI-driven stock recommendation system requires broad understanding of the latest breakthroughs in machine learning, financial analysis, and AI-based recommendation systems. The subsequent literature review compiles findings from 15 technical articles that have appeared since 2020. These comprise a basis for the methods and approaches applied in this research.

A systematic review by [1] explores advancements in AI-based recommender systems, emphasizing the importance of ethical considerations in their development. The study highlights the need for transparency and fairness, which are crucial in financial applications where recommendations can significantly impact investment decisions.

Priel and Rokach [2] proposed a machine learning approach to stock selection, combining value investing and quality features. Their methodology shows the capabilities of machine learning in reinforcing traditional investment strategies by making the right identification of underpriced stocks.

Zhang, Lu, and Jin [3] discussed the implementation of artificial intelligence in the context of recommender systems based on computational intelligence and machine learning algorithms. Their contribution has provided insight into prediction accuracy improvement, which could be applied to financial recommendation systems.

A paper by [4] presents a machine learning-based value investing model that learns investment strategies using complex data pipelines. The study establishes the strength of machine learning in the replication and extension of traditional stock-picking techniques.

A review by [5] focuses on AI techniques for stock price prediction. The paper reviews research conducted between 2020 and 2023. The paper provides a systematic review of the applications of AI in financial markets, detailing the progression of predictive models.

The SPCM proposed by [6] is based on the idea of combining sentiment analysis and price data to improve stock recommendation systems. This approach addresses challenges such as information cocoons, offering a more holistic view of stock evaluation.

Gu, Kelly, and Xiu [7] provide a comparative analysis of the methods of machine learning in empirical asset pricing. Their analysis proves that the methods used here are better suited to measuring asset risk premia and supports the use of machine learning in financial modeling.

Lee, Kim, and Lee [8] developed a temporal graph network with diversification-enhancing contrastive learning for stock recommendation. Their model successfully captures time-varying collaborative signals to enhance recommendation accuracy.

Zhang et al. [9] design StockAgent-a multi-agent AI that represents the investor trading behaviors of the real world environment. This study illustrates large language models' use in finance for decision-making processes.

Study on Next-generation personalized investment recommendation systems- Next-generation personal investment recommender systems discuss data processing and model training and testing/evaluation [10]. It provides valuable insights toward AI-driven financial advisory tool design.

A literature review by [11] presents machine learning applications in business and finance, which ranges from marketing to stock analysis and demand forecasting. The author emphasizes the increasing importance of deep learning techniques in financial settings.

The DeepValue framework by [12] is a deep learning application to value-based investment strategies, which can be used as a comparable framework for company valuation. This is similar to the principles of value investing.

A review by [13] looks at the use of artificial intelligence in stock market prediction, which focuses on neural and hybrid-neuro techniques. It classifies different AI methods that are used in financial forecasting.

Research by [14] discusses the integration of machine learning into value investing in credits. The next frontier in financial analysis is discussed by the study. The study delves into how AI can be used to enhance traditional credit investing strategies.

Shen, Yuan, and Jin [15] present AlphaMLDigger, a machine learning approach for the exploration of excess returns on investment. The authors' work centered on the mining of appropriate information for investment decisions amid volatile markets.

The general theme that emerges from reviewed literature is the increasing role of Artificial Intelligence (AI) and machine learning applications in financial markets, notably in the area of recommending stocks. State-of-the-art techniques such as XGBoost, deep learning frameworks, and ensemble techniques have promised to handle imbalanced data, extract complex patterns, and improve prediction accuracy. However, most existing systems are focused on price predictions, sentiment analysis, or market trends without deeply integrating fundamental value-based metrics like Return on Equity or Debt-to-Equity ratio as core drivers of decision-making. Moreover, although models such as ensemble classifiers and graph neural networks improve the accuracy, they are less adopted in sensitive fields such as finance due to their inability to interpret. To overcome these gaps, this paper suggests a new stock recommendation system based on value investing principles that uses financial fundamentals instead of short-term market trends. The system has strong preprocessing, automated hyperparameter tuning using Optuna, and a varied set of machine learning models including Random Forest, XGBoost, and Stacking Classifiers. A strong evaluation framework through metrics such as F1-Score and AUC-ROC provides reliable and robust predictions. In addition, the best performing model deployed through a web application enhances accessibility for real-world decision-making, thereby closing the gap between advanced AI methods and their practical usability for retail and institutional investors.

III. Dataset Description

The dataset used for this project was downloaded from Ticker Tape. It contains complete data on Indian stocks and has 4,245 records with 31 attributes, including all the important financial metrics, market trends, and sectoral benchmarks that are needed to judge the intrinsic value of a company. The attributes include financial ratios such as Return on Equity (ROE), Net Profit Margin, and Debt-to-Equity, valuation metrics including PE Ratio and PB Ratio, and sector benchmarks that include Sector PE and Sector PB. The dataset includes investment insights that include a change in the Mutual Fund (MF), Foreign Institutional Investors (FII), and Domestic Institutional Investors (DII) holdings for 3 months and 6 months along with earnings data like EPS and Dividend Per Share. These altogether give an entire view of the health and market position of the company. The target variable, R2, categorizes the stocks into three categories as follows: 0-not recommended, 1-recommended, and 2-highly recommended. There is a slight class imbalance of 41% as ”highly recommended,” 31% as ”recommended,” and 28% as ”not recommended.” Preprocessing involved imputation of missing values, Winsorization for outlier handling, and scaling/encoding to make the model compatible. This dataset is a great base for developing an AI-based stock recommendation system that employs machine learning algorithms to classify stocks and help investors in making data-driven decisions.

IV. Methodology

The methodology for developing this stock recommendation system incorporates fundamental principles of artificial intelligence, machine learning, and deployment of software. The AI agent within this project acts as an intelligent system, processing raw financial data that is in the nature of percepts and thus accomplishes the desired state of goal achievement through its function of stock classification to result in accurate recommendations as shown in the Fig 1. Following are the subsections for a comprehensive overview of methodology, with innovative approaches that include pipeline integration, hyperparameter tuning, pruning, and system deployment.

A. Data Collection and Labeling

The stock recommendation system is modeled as the AI agent acting to interact with its environment-the Indian stock market data-to perform intelligent decision-making. The key components of the agent are as follows :

1) Start State: The preprocessing step is initiated with the raw data extracted from Ticker Tape, consisting of 4,245 records and 31 attributes. It encompasses financial ratios, valuation metrics, sector benchmarks, and investment activity indicators prepared and fine-tuned for analysis with data cleaning, handling missing values, and transformation.

2) Percepts: Observations from the data set, such as return on equity, EBITDA margin, PE ratio, debt-to-equity ratio, and market capitalization-which the AI agent uses in order to make predictions

3) Goal State: Classifying stocks based on value investing principles; that is, 0 (not recommended), 1 (recommended), and 2 (highly recommended)

4) Actuators: The machine learning models (which are described below) transform processed data into actionable predictions.

This agent-based approach ensures systematic interaction with data while it progresses from the raw input state to actionable stock classifications.

B. Data Collection and Preprocessing

Ticker Tape’s dataset was used as the agent’s perceptual input, offering detailed insights into financial metrics and stock characteristics. Data preprocessing involved below

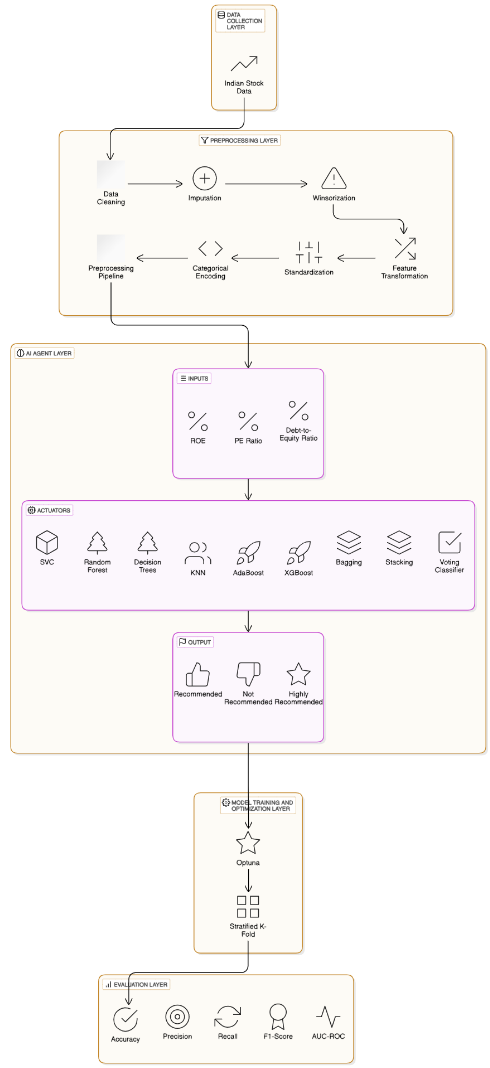

Fig. 1. System Architecture

1) Treatment of Missing Values: Continuous feature missing values are imputed with mean. Other particular variables, namely MF Holding Change and FII Holding Change were assigned 0 if their respective values are missing.

2) Outlier Treatment with Winsorization: To decrease the impact of extreme values, Winsorization was performed on financial metrics such as PE Ratio and ROCE by capping outliers within a reasonable range.

3) Features Transformation: Numerical Features: All numerical features are standardized using StandardScaler. Category features, such as Sector, have been encoded using One-Hot Encoding since such models do not support categorical variables.

4) Pipeline Integration: A novel feature of this approach is that a pipeline for the data preprocessing was designed to achieve automation, thus ensuring consistency and efficiency in cross-validation and model evaluation.

C. Model Selection and Training

An AI agent was developed with three possible classes for the stock. This was to be based on a wide array of machine learning models and therefore to act as actuators of processing data into actionable recommendations. Among them, an ensemble method using several decision trees resulted in the Random Forest algorithm. Similarly, the Support Vector Classifier (SVC) was used with linear as well as non-linear kernels in order to capture rich relationships and patterns of the data involved. Decision Trees, although very simple and interpretable, were also included to offer a clear, hierarchical decision-making structure. K-Nearest Neighbors algorithm, for more flexible and non-parametric classification, relies on the similarity between data points to base its predictions. Boosting techniques like AdaBoost and XGBoost were highly effective for dealing with imbalanced data and high-dimensional features. These methods improved over time by building up weaker models and delivered strong performance with high precision in classification. In an attempt to strengthen the resilience of the AI agent, ensemble techniques including Bagging and the Voting Classifier were implemented. Bagging reduced the variance by combining several iterations of the decision trees while the Voting Classifier aggregated predictions from various models in achieving consensus with either hard or soft voting. Lastly, the Stacking Classifier pushed ensemble learning to a new level by combining base classifiers' predictions into a meta-classifier, thus increasing performance through the strength of the individual models. All the models were applied inside pipelines that automated preprocessing, training, and evaluation, thus streamlining the workflow for the project throughout.

D. Hyperparameter Tuning

A crucial innovation in this methodology was the utilization of Optuna, which is an automated hyperparameter optimization framework, for fine-tuning models. The following parameters were optimized. Pruning was used in Optuna experiments to end poorly performing configurations early, thus saving computation and speeding up the optimization process.

E. Evaluation Metrics

The performance of each model was thoroughly tested using a comprehensive set of metrics in order to ensure robustness and reliability. Accuracy was the most important indicator of model performance since it simply measures the overall proportion of correctly classified stocks. However, as the imbalanced nature of the data classes slightly, more metrics: Precision, Recall, F1-Score were further computed for each class to detail even further the way the different models dealt with imbalanced classes. The metrics calculated allow a deeper view and therefore can offer a measure for minimizing false positives versus false negatives for all different categories of stocks. The Area Under the Receiver Operating Characteristic Curve, AUC-ROC was used as an important measure of evaluation, determining whether different models could distinguish among different categories of stocks. This metric was calculated both as macro-average and per-class, which showed which models were better suited to predict stocks as ”Not Recommended,” ”Recommended,” or ”Highly Recommended.” Higher AUC-ROC scores reflected higher quality classification, especially when a scenario called for differentiation among three classes to make actionable investment decisions. Stratified K-Fold Cross-Validation was used to ensure that the evaluation was not biased and reliable. It divided the dataset into three folds such that the proportion of classes in the training and validation sets was the same. This reduces the possibilities of overfitting and ensures that models generalize well to different subsets of data. Combined with stratified cross-validation, these evaluation metrics gave a robust framework for selecting the best-performing models for recommending stocks.

F. Deployment

The deployment phase of this project focused on making the best-performing model, XGBoost, accessible and interactive. The reason behind choosing XGBoost for deployment was its superior performance across evaluation metrics, including accuracy, F1-Score, and AUC-ROC. For smooth integration and usability, the trained XGBoost model and the preprocessing pipeline were serialized using Pickle, allowing efficient reusability without requiring the retraining of the model. This serialization ensured that the whole workflow of data preprocessing and model inference could be executed seamlessly. A Flask-based web application was developed to make it accessible to end users; the stock recommendation system consists of a web interface and a user-friendly form on which the users can input any financial metrics of a given stock, such as the Return on Equity, PE Ratio, and Debt-to-Equity Ratio. The Flask application loads the Pickle file that contains both the preprocessing pipeline and the trained XGBoost model, processes user inputs in real-time, and predicts the category of the stock as ”Not Recommended,” ”Recommended,” or ”Highly Recommended.” It presents the predictions clearly and concisely, thereby enabling users to make the best decisions. This is a deployment approach that fills the gap between advanced AI-driven analysis and practical usability. By enabling real-time predictions through a web interface, the AI agent becomes a very powerful tool for assisting investors in making informed decisions based on value investing principles. It demonstrates how advanced machine learning models can be effectively operationalized for real-world applications, providing an intuitive and accessible solution for stock recommendation.

V. Result and Analysis

This section provides an in-depth comparison of the performance of various machine learning models before and after using Optuna to tune hyperparameters for a stock recommendation system. The evaluation is presented in multiple subsections based on overall performance, specific model results, and some insights from experimentation.

A. Overall Model Performance

The project classified the stock into three categories: ”Not Recommended,” ”Recommended,” and ”Highly Recommended,” using nine machine learning models. These included Random Forest, Support Vector Classifier, Decision Tree, AdaBoost, K-Nearest Neighbors, XGBoost, Bagging, Voting Classifier, and Stacking Classifier. Many main metrics, such as accuracy, macro precision, macro recall, macro F1-score, and macro AUC-ROC scores, measure the performance of the models. These figures suggest that most of these models have improved with much more significant margins after this hyperparameter tuning, that is, accuracy and also AUC-ROC, indicating that optimization is also an important step to strengthen the predictability.

B. Performance Before Hyperparameter Tuning

Prior to optimization, the XGBoost model had the best performance with an accuracy of 86.71%, F1-Score of 86.66%, and a macro AUC-ROC of 96.2% as shown in Table 1 and Table 2. Other models like Random Forest and AdaBoost also performed very well with accuracies of 78.25% and 79.71%, respectively. The simpler models like SVC performed very poorly with an accuracy of 41.06% and an F1-Score of 19.50% as shown in Table 3 and Table 4. Although ensemble techniques- Voting Classifier and Stacking Classifier-showed a passable performance with accuracy reaching about 77%, an area for improvement of ensemble ability in handling imbalanced data can clearly be seen by the generally lower recall and AUC- ROC scores of a particular class. Performance After Hyperparameter Tuning The main im- provements observed after tuning of hyperparameters using Optuna were seen for all the models, as indicated in the following table :

TABLE I
PERFORMANCE METRICS OF MODELS BEFORE HYPERPARAMETER TUNING

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1 Score (Macro)
ada_boost	0.8111	0.8107	0.8077	0.8079
bagging	0.8912	0.9024	0.8918	0.8927
svc	0.5524	0.5542	0.5081	0.4544
random_forest	0.8938	0.9010	0.8912	0.8938
knn	0.7086	0.7200	0.7132	0.7129
voting	0.8002	0.8057	0.8012	0.8030
xgboost	0.9048	0.9117	0.9028	0.9046
decision_tree	0.8919	0.8893	0.8906	0.8897
stacking	0.8872	0.8986	0.8866	0.8889

C. Insights from AUC-ROC Scores

The AUC-ROC (Macro) scores capture an overall assessment of how the models would distinguish between the three stock classes: ”Not Recommended,” ”Recommended,” and ”Highly Recommended.” The evaluation measures the good practice with which the models separate classes irrespective of any imbalance in the dataset. With the models after tuning the hyperparameters, the majority obtained an AUC-ROC (Macro) score higher than 90%, implying a good handling of

TABLE II
AUC-ROC METRICS OF MODELS BEFORE HYPERPARAMETER TUNING

Model	AUC-ROC Class 0	AUC-ROC Class 1	AUC-ROC Class 2
ada_boost	0.9639	0.9190	0.8988
bagging	0.9695	0.9616	0.9758
svc	0.9175	0.8122	0.8204
random_forest	0.9682	0.9653	0.9754
knn	0.9236	0.8415	0.8626
voting	0.9585	0.9101	0.9208
xgboost	0.9716	0.9707	0.9813
decision_tree	0.9212	0.9240	0.9424
stacking	0.9527	0.9485	0.9705

TABLE III
PERFORMANCE METRICS OF MODELS BEFORE HYPERPARAMETER TUNING

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1 Score (Macro)
ada_boost	0.8111	0.8107	0.8077	0.8079
bagging	0.8912	0.9024	0.8918	0.8927
svc	0.5524	0.5542	0.5081	0.4544
random_forest	0.8938	0.9010	0.8912	0.8938
knn	0.7086	0.7200	0.7132	0.7129
voting	0.8002	0.8057	0.8012	0.8030
xgboost	0.9048	0.9117	0.9028	0.9046
decision_tree	0.8919	0.8893	0.8906	0.8897
stacking	0.8872	0.8986	0.8866	0.8889

complexities with multi-class classification tasks. Models such as XGBoost, Random Forest, and Bagging showed consistent good performance; their macro-level scores are reflective of good classification in all categories. When analyzing class-specific AUC-ROC scores, Class 0 (Not Recommended) was always the highest in nearly all models. This means that the models were very good at identifying stocks with poor financial metrics and low investment potential. Class 2 (Highly Recommended) was slightly lower in some of the models, such as KNN and SVC, which means that distinguishing the best-performing stocks needed more refinement. This may be due to the fact that high-performance stocks and those in the ”Recommended” category have some overlapping characteristics, which is complex, and therefore, the advanced models and further feature engineering are required to increase the

TABLE IV
AUC-ROC METRICS OF MODELS BEFORE HYPERPARAMETER TUNING

Model	AUC-ROC Class 0	AUC-ROC Class 1	AUC-ROC Class 2
ada_boost	0.9639	0.9190	0.8988
bagging	0.9695	0.9616	0.9758
svc	0.9175	0.8122	0.8204
random_forest	0.9682	0.9653	0.9754
knn	0.9236	0.8415	0.8626
voting	0.9585	0.9101	0.9208
xgboost	0.9716	0.9707	0.9813
decision_tree	0.9212	0.9240	0.9424
stacking	0.9527	0.9485	0.9705

precision in this class.

D. Impact of Hyperparameter Tuning

The crucial role that hyperparameter tuning plays in optimizing key parameters is also tailored to the algorithm applied. In Random Forest, for instance, tuning estimators, maximum depth, and minimum samples split would significantly improve the ability to capture complex patterns without being overfit. Similarly, the best model of all, XGBoost, was also fine-tuned on parameters such as learning rate, number of estimators, and subsample ratio, which balanced exploration and exploitation in the gradient-boosting iterations of XGBoost and resulted in much better classification accuracy and strong AUC-ROC scores. Other techniques, like Bagging, show superior performance when the number of estimators and the depth of base learners are optimized for them to make it more resistant to variance in the dataset. With the Support Vector Classifier (SVC), fine-tuning the regularization parameter, C and the type of kernel resulted in a better performance at dealing with non-linear data, although the improvement is more modest compared to the ensemble methods. These optimizations ensured that each model was properly configured to address the specific characteristics of the dataset, including class imbalance and overlapping financial metrics. Pruning of configurations during Optuna trials was an innovative tuning technique: it terminated underperforming configurations early, significantly reduced the amount of computational overhead needed for the process, and stayed focused on promising combinations of parameters. Pruning, in this way, helped streamline the optimization procedure towards rapid convergence to models performing well, ensuring high computation efficiency while achieving good performance.

E. Performance After Hyperparameter Tuning

The main improvements observed after tuning of hyperparameters using Optuna were seen for all the models, as indicated in the following table :

1) Stacking Classifier: Performed better by achieving an accuracy of 88.71% with an F1-Score of 88.89%, and a macro AUC-ROC of 95.72%.Other models such as KNN and SVC showed minor improvements, though they were behind the ensemble methods. The accuracy obtained by KNN was 70.85% while that of SVC was 55.24%. The simplicity of these models resulted in an inability to capture the richness in the data as against more advanced ensemble and boosting techniques.

2) Random Forest: All-around improvement with the rise to 89.37% accuracy, 89.38% for the F1-Score, and the macro AUC-ROC rising to 96.96%. This signifies that improved hyperparameters like number estimators and the depth did have a substantial difference as shown in Fig 2.

3) Bagging Classifier: Aattained accuracy at 89.11% and also with macro AUC-ROC of 96.89% .It also has high recall as shown in Fig. 3.

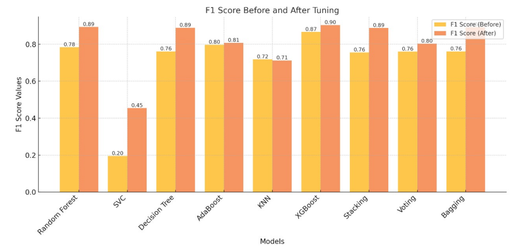

Fig. 2. F1 Score Before and After Tuning

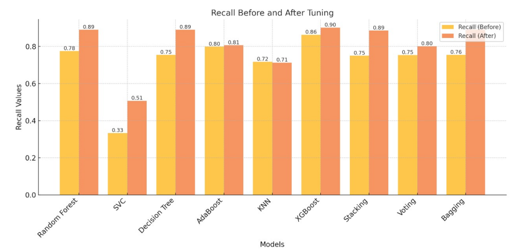

Fig. 3. Recall Before and After Tuning

4) XGBoost: Stands the best, but improved significantly with 90.48% accuracy, 90.46% F1-Score and high precision as shown in Fig. 4 and 5, and the macro AUC-ROC of 97.45%. Class-specific AUC-ROC for XGBoost also stands out, standing at 97.16% for Class 0, 97.06% for Class 1, and 98.13% for Class 2.

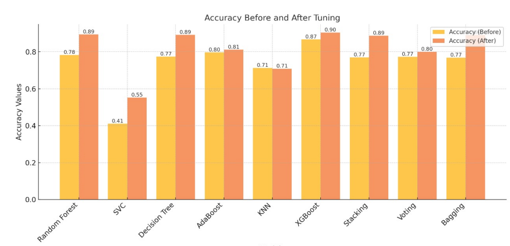

Fig. 4. Accuracy before and after tuning

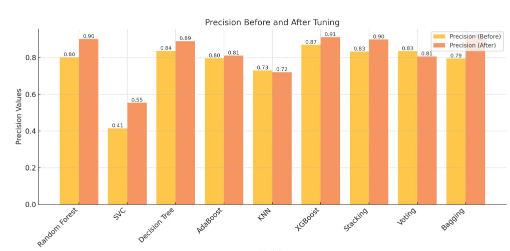

Fig. 5. Precision before and after tuning

F. Comparison of Ensemble Models

Ensemble methods such as Voting Classifier, Bagging, and Stacking outperformed individual models because it can combine the strengths of multiple base learners. However, in the Stacking Classifier, using XGBoost as the meta-classifier produced particularly strong results in terms of high accuracy and AUC-ROC scores that closely matched the performance of XGBoost and Random Forest.

G. Deployment-Ready Model

Based on the result, XGBoost was considered to be the best for deployment. Its high accuracy, F1-Score, and macro AUC-ROC ensure the correct classification of stocks. Using Pickle, the model along with the preprocessing pipeline was serialized and deployed on a Flask-based web application for real-time predictions.

VI. Acknowledgment

We would like to express our heartfelt gratitude to all those who contributed to the completion of this research project on the Stock Selection Using Machine Learning and Value Investing Strategies. We would like to thank our institution Amrita Vishwa Vidyapeetham for providing access to resources and data that facilitated our research. Their contributions were crucial to the success of this project.

VII. Conclusion

This paper represents a strong, comprehensive strategy for developing the stock recommendation system using the principles of artificial intelligence and machine learning. This system then classifies the stock in three actionable categories: namely "Not Recommended," "Recommended," and "Highly Recommended." Diversity in Machine Learning models - ensemble techniques such as XGBoost and Bagging with Stacking have been used with hyperparameter fine-tuning using Optuna - was much more apparent in improvements in terms of predictive performance. The XGBoost model was the best performing one, as it achieved the highest accuracy, F1-Score, and macro AUC-ROC scores, making it an ideal candidate for deployment. Smooth interaction was allowed in a Flask-based web application by deploying the model, where users could input financial metrics and get real-time recommendations. This study demonstrates how AI can bridge the divide between traditional value investing and modern computational techniques, presenting a reliable and user-friendly solution for stock selection in the Indian market.

References

[1] Masciari, Elio, Areeba Umair, and Muhammad Habib Ullah. "A Systematic Literature Review on AI based Recommendation Systems and their Ethical Considerations." IEEE Access 2024.

[2] Priel, Ronen, and Lior Rokach. "Machine learning-based stock picking using value investing and quality features." Neural Computing and Applications 2024: 1-24.

[3] Priel, Ronen, and Lior Rokach. "Machine learning-based stock picking using value investing and quality features." Neural Computing and Applications 2024: 1-24.

[4] He, Jun Yi Derek, and Joseph Ewbank. "Development and Evaluation of a Machine Learning-Based Value Investing Methodology." Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020. Springer International Publishing, 2021.

[5] He, Jun Yi Derek, and Joseph Ewbank. "Development and Evaluation of a Machine Learning-Based Value Investing Methodology." Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020. Springer International Publishing, 2021.

[6] Wang, Jiawei, and Zhen Chen. "SPCM: A Machine Learning Approach for Sentiment-Based Stock Recommendation System." IEEE Access 2024.

[7] Gu, Shihao, Bryan Kelly, and Dacheng Xiu. "Empirical asset pricing via machine learning." The Review of Financial Studies 33.5 2020: 2223-2273.

[8] Lee, Youngbin, Yejin Kim, and Yongjae Lee. "Stock Recommendations for Individual Investors: A Temporal Graph Network Approach with Diversification-Enhancing Contrastive Learning." arXiv preprint arXiv:2404.07223 2024.

[9] Zhang, Chong, et al. "When ai meets finance (stockagent): Large language model-based stock trading in simulated real-world environments." arXiv preprint arXiv:2407.18957 2024.

[10] Xue, Jingming, et al. "A bi-directional evolution algorithm for financial recommendation model." Theoretical Computer Science: 35th National Conference, NCTCS 2017, Wuhan, China, October 14-15, 2017, Proceedings. Springer Singapore, 2017.

[11] Nazareth, Noella, and Yeruva Venkata Ramana Reddy. "Financial applications of machine learning: A literature review." Expert Systems with Applications 219 2023: 119640.

[12] Huang, K. J. "DeepValue: a comparable framework for value-based strategy by machine learning." Computational Economics 60.1 2022: 325-346.

[13] Ferreira, Fernando GDC, Amir H. Gandomi, and Rodrigo TN Cardoso. "Artificial intelligence applied to stock market trading: a review." IEEE Access 9 2021: 30898-30917.

[14] Belhadi, Amine, et al. "An ensemble machine learning approach for forecasting credit risk of agricultural SMEs' investments in agriculture 4.0 through supply chain finance." Annals of Operations Research 2021: 1-29.

[15] Shen, Q., Yuan, W., & Jin, Y. (2024). AlphaMLDigger: Machine Learning for Excess Return Exploration in Investment Strategies. Journal of Financial Analytics.