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
k_sreelakshmi@blr.amrita.edu

***Abstract*—Research identifies an AI-based stock recommendation system based on value-investing principles. Stock classification will be categorized into "Not Recommended," "Recommended," and "Highly Recommended." The dataset will be derived from Ticker Tape, along with detailed ratios of financials, market trend, and sectoral benchmark. Preprocessing techniques to be applied include Winsorization, imputation, and feature scaling to fit the data for machine learning. Three models—XGBoost, Random Forest, and Decision Tree—were tested based on metrics such as accuracy, F1-score, and AUC-ROC. Hyperparameter tuning with Optuna significantly improved the model's performance, and the best model was XGBoost, which achieved 91.99% accuracy in testing and an AUC-ROC of 98.59%. The system is implemented as a Flask-based web application, which uses the XGBoost model and preprocessing pipeline to make real-time stock predictions. This project will show how AI, in advanced models and optimized techniques, can help investors make value investment 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 is transforming industries by mimicking human cognitive functions such as decision-making, learning, and problem-solving. It has emerged as a revolutionary tool in the financial sector, enabling advanced stock analysis and recommendation systems. Using extensive datasets and computational power, AI reveals insights previously inaccessible. This paper develops an AI-based stock recommendation system based on value investing principles that focus on investment in long-term basis. It navigates an ever-complex environment, working through large amounts of financial data and providing the basis for investor recommendations. It consists of an agent-based architecture as the main core that operates as a decision-making unit. Such an intelligent agent analyzes large sets of financial metrics - historical performance, market trends, and also fundamental ones such as ROE, EBITDA Margin, and Debt-to-Equity Ratio. These metrics are referred to as percepts - the observable parts of the environment that are captured by an agent in order to make some reasoned judgments regarding the stock market. It seeks to classify stocks into three groups: Not Recommended, Recommended, and Highly Recommended. Classification according to value investing will enable investors to determine those undervalued stocks that are likely to experience rapid growth. This AI journey starts at a raw dataset obtained from the Ticker Tape and gives it sector-specific insights in the form of financial metrics, market trends, amongst other things. The steps ensure that the data being worked upon is robust to handle analysis because it has been preprocessed through Winsorization in order to handle outliers and also imputes missing values with the technique mean and KNN Imputation. The goal state of the AI system would be to generate accurate and actionable stock recommendations, thereby helping investors make informed decisions. This is accomplished by using actuators in the form of machine learning algorithms, which are Random Forest, Decision Tree, and XGBoost. These models operate on the preprocessed data for the classification of stocks regarding their value-driven investment. In addition, advanced hyperparameter optimization techniques, such as Optuna, are used to fine-tune model parameters in order to enhance predictive accuracy and decision-making efficiency. The optimization process ensures the effective adaptation of models towards the complexities of financial data. This paper describes the design, development, and evaluation of this AI-driven stock recommendation system. It bridges the gap between traditional value investing strategies and cutting-edge computational methods by integrating them. It provides a scalable and robust solution for stock selection in the Indian market, empowering investors with data-driven insights for long-term wealth creation.

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 as shown in the Fig 1. 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. 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

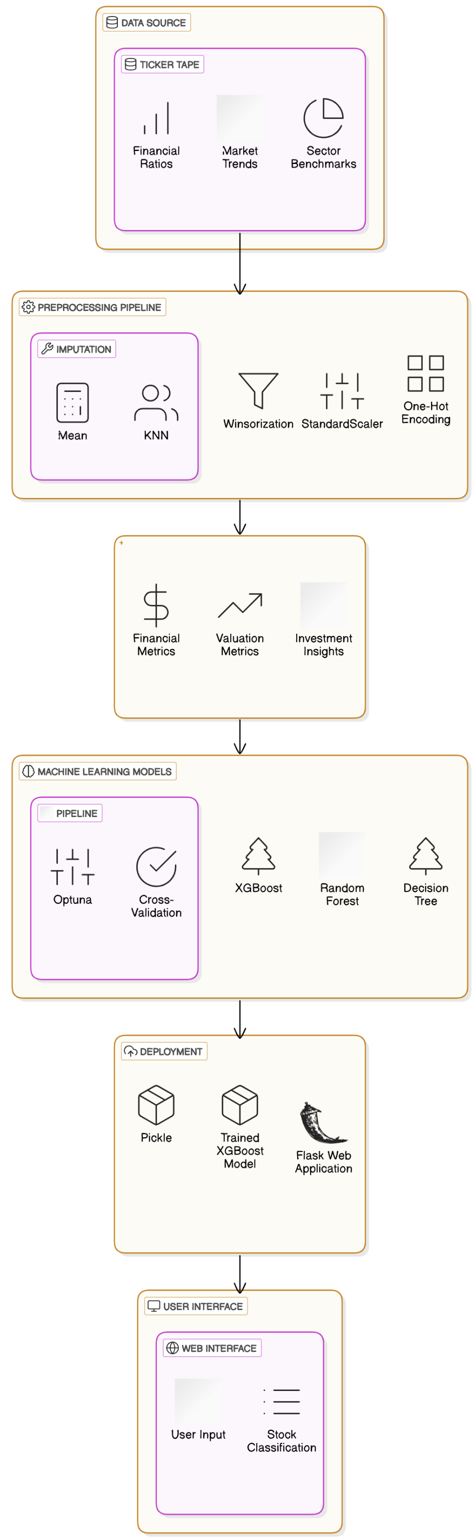

Fig. 1. System Architecture

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 for three possible categories of classification, using a robust array of machine learning models as actuators by processing financial data into actionable recommendations. In these models, Random Forest, Decision Tree, and XGBoost were the models that best suited and proved to be effective with the handling of financial data in order to deliver high-performance classification. The Random Forest algorithm, which is an ensemble method using multiple decision trees, was one of the standout performers. It showed robustness, reduced variance, and the ability to handle complex relationships in the data by aggregating the predictions of several decision trees. Its ability to maintain balanced class predictions made it particularly suitable for this project. Decision Trees were included for their simplicity and interpretability, providing a hierarchical, rule-based approach to decision-making. They gave clear insights into how different financial metrics affected the classification process, which is essential for explainability in stock recommendations. This proved to be highly effective with imbalanced data and a high-dimensional feature in terms of a gradient boosting algorithm. The iterative method to build and correct the weaker models was instrumental for obtaining high precision and great performance. Its capability of both optimizing speed and accuracy led to being a necessary element of the framework used in the AI agent. Pipelined to automate the preprocessing, training, and evaluation, the three models were integrated in that pipeline, ensuring a uniform application of data transformations and parameter tuning. This was done by focusing on these three algorithms that had great performance as a way to balance the trade-off of simplicity with interpretability and predictive power.

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.

TABLE I
HYPERPARAMETERS FOR MODELS

Model	**Parameters**
Random Forest	`n_estimators`, `max_depth`, `min_samples_split`, `class_weight`, `random_state`, `n_jobs`
Decision Tree	`max_depth`, `criterion`, `min_samples_split`, `class_weight`, `random_state`
XGBoost	`n_estimators`, `max_depth`, `learning_rate`, `subsample`, `colsample_bytree`, `eval_metric`, `random_state`, `n_jobs`

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

The performance of the stock recommendation system was tested by running three models, namely Random Forest, Decision Tree, and XGBoost, on both the training and testing datasets. Accuracy, precision, recall, F1 score, and AUC-ROC metrics were considered for overfitting and underfitting along with hyperparameter optimization using Optuna. For real-time prediction, the XGBoost model was deployed via a Flask-based web application.

A. Models

1) Random Forest: The training and testing accuracies of the model were 94.76% and 87.04%, respectively, reflecting a performance drop of roughly 7.7% for the training to the testing dataset. Training AUC-ROC was very high at 99.59%, but testing AUC-ROC dropped to 96.43%, pointing towards slight overfitting. This is within the nature of Random Forest as an ensemble model, handling variance appropriately but overfitting when the hyperparameters are not well-tuned. Applying hyperparameter optimization using Optuna finally gave training accuracy to the level of 99.94% and improved the testing accuracy up to 89.28%. Yet, the gap between the training and testing accuracies went even wider that indicates that there was an increase in overfitting due to optimization. AUC-ROC testing data improves marginally up to 97.48%, though at the cost of diminished generalization. The Random Forest model is strong in prediction but prone to overfitting, even more so after optimization. This translates into much lower generalizability compared to XGBoost and may thus require further fine-tuning for optimal blending of training and testing.

2) Decision Tree: Decision Tree showed a train accuracy of 88.93% and a test accuracy of 86.21%, with a drop between the two datasets of merely 2.72%. This shows that the decision tree was much less sensitive to overfitting issues than Random Forest. And its overall accuracy and AUC-ROC were really low, with the former being at 95.52% in the case of testing AUC-ROC, which shows this model has a limited potential to capture complex patterns data. The hyperparameter optimization indeed greatly improved the training accuracy to 98.06% whereas the testing accuracy marginally improved to 88.57%. The AUC-ROC for the test data also marginally improved at 93.15%. Still, the Decision Tree began showing signs of overfitting after the hyperparameter optimization process as there is a much larger gap in training and testing accuracies. Decision Tree is quite simple and interpretable; hence, it is applicable for fast decision-making but has a lower predictive power than ensemble methods. The optimization has improved its performance, but the model remains less robust than Random Forest and XGBoost.

3) XGBoost: With the training accuracy at 93.23% and the testing accuracy at 90.81%, the gap between them was minimal, being just 2.42%. This shows that it had a strong generalization capability. The AUC-ROC metrics were uniformly high with the testing AUC-ROC at 98.26%, which shows XGBoost's ability to deal with class imbalance and complex relationships. With hyperparameter optimization, XGBoost reached 100% training accuracy and improved the testing accuracy to 91.99%, the highest value among all models. Also, the AUC-ROC metrics have slightly improved, with a testing AUC-ROC of 98.59%. Notably, the gap between training and testing performance remains small for XGBoost, which makes it very robust and has good generalization capabilities even after optimization. XGBoost outperformed all other models in the pre- and post-optimization scenarios. This gradient boosting framework is very effective in correcting errors iteratively and therefore suitable for high-dimensional and imbalanced data. The minimal overfitting that was even found after optimization makes it a good performer.

TABLE II
MODEL PERFORMANCE ON TEST DATA BEFORE TUNING

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1 Score (Macro)
random_forest	0.870435807	0.880499297	0.874744666	0.875157351
decision_tree	0.862190813	0.884302421	0.872387858	0.867932465
xgboost	0.908127208	0.917320655	0.909471518	0.909721798

TABLE III
MODEL PERFORMANCE ON TEST DATA BEFORE TUNING (AUC-ROC SCORES)

Model	**AUC-ROC (Macro)**	**Not Recommended**	**Recommended**	**Highly Recommended**
random_forest	0.964304451	0.972661542	0.956627169	0.963624642
decision_tree	0.955216765	0.969716869	0.940821678	0.955111748
xgboost	0.982597536	0.983742085	0.981047656	0.983002865

TABLE IV
MODEL PERFORMANCE ON TEST DATA AFTER TUNING OPTUNA

Model	**Accuracy**	**Precision (Macro)**	**Recall (Macro)**	**F1 Score (Macro)**
random_forest	0.892815077	0.899700446	0.892899663	0.894500353
decision_tree	0.885747939	0.882889132	0.887153184	0.884561001
xgboost	0.919905771	0.926516762	0.917634838	0.920262352

TABLE V
MODEL PERFORMANCE ON TEST DATA AFTER TUNING OPTUNA (AUC-ROC SCORES)

Model	**AUC-ROC (Macro)**	**Not Recommended**	**Recommended**	**Highly Recommended**
random_forest	0.97484618	0.977088921	0.971882284	0.975567335
decision_tree	0.93153503	0.933032875	0.925216913	0.936535301
xgboost	0.985856537	0.98602317	0.985236985	0.986309456

TABLE VI
MODEL PERFORMANCE ON TRAIN DATA BEFORE TUNING

Model	**Accuracy**	**Precision (Macro)**	**Recall (Macro)**	**F1 Score (Macro)**
random_forest	0.947585395	0.950777736	0.948584446	0.94819044
decision_tree	0.889281508	0.906082672	0.89314527	0.89190913
xgboost	0.932273263	0.938331101	0.929463156	0.931160328

TABLE VII
MODEL PERFORMANCE ON TRAIN DATA BEFORE TUNING (AUC-ROC SCORES)

Model	**AUC-ROC (Macro)**	**Not Recommended**	**Recommended**	**Highly Recommended**
random_forest	0.995870596	0.998420089	0.993840566	0.995351134
decision_tree	0.971613373	0.978937118	0.959899806	0.976003193
xgboost	0.994771695	0.99674967	0.993081957	0.994483458

TABLE VIII
MODEL PERFORMANCE ON TRAIN DATA AFTER TUNING

Model	**Accuracy**	**Precision (Macro)**	**Recall (Macro)**	**F1 Score (Macro)**
random_forest	0.999411072	0.999371069	0.999407773	0.999388987
decision_tree	0.980565371	0.97900345	0.981901525	0.98036271
xgboost	1	1	1	1

TABLE IX
MODEL PERFORMANCE ON TRAIN DATA AFTER TUNING (AUC-ROC SCORES)

Model	**AUC-ROC (Macro)**	**Not Recommended**	**Recommended**	**Highly Recommended**
random_forest	0.999999881	1	1	0.999999642
decision_tree	0.999185384	0.999763683	0.998739693	0.999052776
xgboost	1	1	1	1

B. Model Performance

Among the three models tested, Random Forest had good predictive power but overfitting issues, particularly after hyperparameter tuning. Decision Tree was simple and interpretable but had lower overall accuracy and less ability to generalize to complex datasets. XGBoost proved to be the best model with high accuracy, robust generalization, and minimal overfitting, which makes it the best choice for this stock recommendation system. Performance compare of the models as shown in Fig 2, Fig 3, Fig 4 and Fig 5.

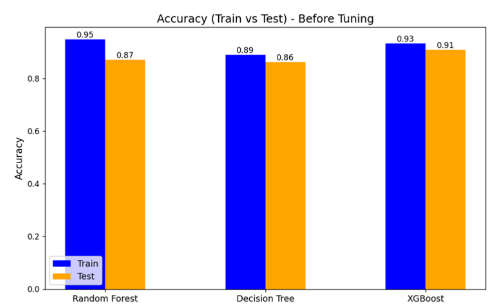

Fig. 2. Accuracy Before Tuning

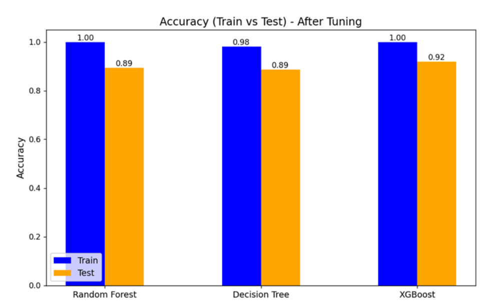

Fig. 3. Accuracy After Tuning

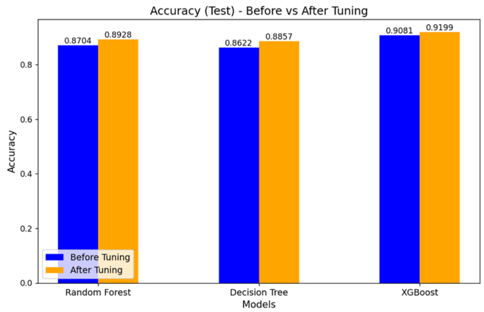

Fig. 4. Accuracy After vs Before Tuning

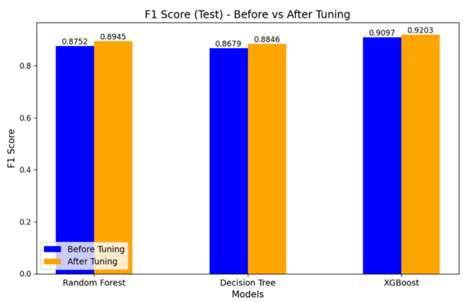

Fig. 5. F1 Score After vs Before Tuning

C. Deployment

With Pickle serialization, the XGBoost model, including its preprocessing pipeline, was deployed on a Flask-based web application to make real-time stock recommendations. In this deployment, the system accepts new financial data and provides immediate predictions to users. Preprocessing and prediction integration provides an effortless experience for the user, which makes the system feasible for end-users. The best performing model for the stock recommendation system is the XGBoost model. It generalized well, handled imbalanced data, and maintained excellent performance metrics both before and after optimization. This model, with the combination of modern computational techniques and value investing principles, provides an excellent solution for stock selection in the Indian market.

VI. Acknowledgment

We would like to express our heartfelt gratitude to all those who contributed to the completion of this research project on AI-Driven Stock Recommendation : Optimizing Predictions with Hyperparameter Tuning. 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 such as XGBoost, random forest, decision tree 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.