Journal of Information Systems and Informatics
Vol. 6, No. 1, March 2024 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v6i1.643 Published By DRPM-UBD

EnsembleForge: A Comprehensive Framework for Simplified Training and Deployment of Stacked Ensemble Models in Classification Tasks

Ubong David Essien[1], Godwin Okon Ansa[2], Aloysius Akpanobong[3]

[1,2,3] Department of Computer Science, Akwa Ibom State University, Ikot Akpaden, Mkpat Enin, Akwa Ibom State, Nigeria
Email: [1]ubongessien@aksu.edu.ng, [2]godwinansa@aksu.edu.ng, [3]aloysiusakpanobong@aksu.edu.ng

Abstract

In this work, we introduce EnsembleForge, a versatile framework designed to streamline machine learning experimentation and simplify classification tasks. Leveraging the stacking ensemble method, EnsembleForge offers an intuitive platform built upon the Scikit-learn library. This framework facilitates seamless model implementation and evaluation, supporting both Randomized and Grid Search for hyperparameter optimization. Our experiments with publicly available datasets demonstrate the ease of use and effectiveness of EnsembleForge in experimenting with various algorithms. With its adaptability and innovation, EnsembleForge showcases promising potential to serve as an asset for researchers and practitioners seeking to achieve optimal model performance in their machine learning endeavors.

Keywords: Machine Learning, Stacking technique, Ensemble model, Classification

1. INTRODUCTION

In the realm of machine learning, the integration of ensemble learning and deep learning has emerged as a transformative paradigm, surpassing the capabilities of traditional algorithms [1-7] Ensemble learning, characterized by the amalgamation of diverse base models into a unified framework, stands out for its ability to yield a more potent model that outperforms its individual components. This paradigmatic shift in machine learning is supported by ensemble methods that leverage multiple learning algorithms to achieve predictive performance superior to that of individual algorithms alone [10,15].This research introduces a novel contribution to the field—an advanced stacked ensemble framework tailored to empower researchers, machine learning enthusiasts, and students engaged in classification tasks which may be highly time-consuming and relies heavily on the expertise and experience of the individuals involved, thus affecting its accuracy[5]. Classification is a data mining technique used to predict group membership for data instances [8]. It becomes tedious when starting with a classification problem.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The EnsembleForge framework is designed to facilitate experimentation with ensemble models, the framework seamlessly integrates popular machine learning models, including random forests (RF), support vector machines (SVM), Kth Nearest Neighbor (KNN), and others for various classification problems and is built on scikit-learn module [9]. Noteworthy features of this framework include support for initial hyperparameter tuning values, granting users the flexibility to explore diverse machine learning algorithms and their associated hyperparameters. Furthermore, the framework accommodates the stacking of more than two models, presenting an avenue for enhanced accuracy in classification tasks. This paper elucidates the framework's design, functionality, and its potential impact on advancing the landscape of ensemble learning in machine learning research and application and further encourages more researchers who wishes to join in to have a platform to start experimenting.

The literature review section provides an in-depth exploration of ensemble learning, emphasizing its importance in improving predictive performance across machine learning tasks. It discusses the increasing attention ensemble methods have received and their superiority over individual algorithms. The section then examines various studies and methodologies within ensemble learning, including transfer learning frameworks, and comparative analysis of classification methods. By critically analyzing these works, the literature review seeks to identify gaps, challenges, and opportunities for further research in ensemble learning, laying the groundwork for subsequent discussions and analysis in the research project.

The work by [8] in 2020 highlights the significant attention garnered by ensemble methods in the field of machine learning and data mining. Ensemble methods, which involve combining multiple learning algorithms, have been recognized for their ability to achieve better predictive performance than individual algorithms. The literature suggests that the combination of multiple learning models has demonstrated substantial improvements both theoretically and experimentally. Ensemble learning algorithms are acknowledged as a dominant and cutting-edge approach, finding applications across various real-world problems, including face and emotion recognition, text classification, medical diagnosis, and financial forecasting.

A visual analytics system for layered generalization and visualization is used by [3] to enable ensemble learning through the use of the knowledge generation model StackGenVis. The creation of StackGenVis, a system that helps users manage data instances, choose a set of high-performing and diverse algorithms, measure predictive performance, and dynamically adapt performance metrics are the paper's primary contributions. With the help of this tool, users can select from a variety of models and eliminate models that perform poorly or overpromise, thus simplifying the final stack. The approach taken in the paper is utilizing visualization approaches to help users make decisions when building a stacking ensemble. The

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

tool has been evaluated through interviews with three machine learning experts, demonstrating its potential for widespread use in the field of machine learning.

Research by [1] presents a novel Dynamic Ensemble Learning (DEL) algorithm for designing ensembles of neural networks (NNs). The main contribution of the paper is the development of a DEL algorithm that automatically determines the size of the ensemble, the number of individual NNs, the number of hidden nodes of individual NNs, and different training samples for individual NNs' learning. The algorithm introduces negative correlation learning for diversity and variation of training samples for individual NNs, providing better learning from the whole training samples. The methodology used in the paper involves a constructive strategy for determining the size of the ensemble, the number of individual NNs, the number of hidden nodes of individual NNs, and different training samples for individual NNs' learning. The algorithm is applied to a set of real-world classification problems, demonstrating that DEL produces dynamic NN ensembles of appropriate architecture and diversity, showing good generalization ability.

Research by [13] introduced a stacking-based evolutionary ensemble learning system, "NSGA-II-Stacking," for predicting the onset of Type-2 diabetes mellitus (T2DM) within five years. The main contribution is the development of this system, which utilizes a multi-objective optimization algorithm for base learner selection and a k-nearest neighbor (K-NN) meta-classifier for model combination. The methodology involves using the publicly accessible Pima Indian diabetes (PID) dataset, preprocessing the data, and applying the NSGA-II-Stacking framework. The research demonstrates that the proposed system significantly outperforms several individual machine learning approaches and conventional ensemble approaches, achieving high accuracy, sensitivity, specificity, f-measure, and area under the ROC curve. The results show the potential of the NSGA-II-Stacking system for early detection and prediction of T2DM.

Research by [11] proposed a stacking-based ensemble framework for predicting hypertension risk prospectively. The main contribution of the paper is the development of the Multi-objective Iterative Model Selection (MoItMS) strategy to maximize the accuracy of meta-learners and the diversity of the ensemble model simultaneously. The methodology involves using the National Health and Nutrition Examination Survey (NHANES) dataset, which includes 11,341 patients, to train and test the model. The research demonstrates that the proposed ensemble framework outperforms 13 individual models and ensemble models in terms of precision, recall, accuracy, F1-measure, and AUC. The proposed system achieves the highest accuracy of 76.82%, sensitivity of 53.76%, specificity of 71.13%, f-measure of 61.05%, and area under the ROC curve of 0.84. The paper also focuses on the impact of lifestyle factors on hypertension classification performance and discovers that lifestyle factors can improve the model in

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

distinguishing hypertensive samples. The proposed method can identify people at high risk of hypertension and can be integrated into community health management systems in the future.

Research by [4] proposes a stacking-based ensemble learning system called SELF for breast cancer classification. The main contribution of the paper is the development of the SELF framework, which combines various machine learning algorithms, such as SVM, k-nearest neighbors, Naive Bayes, and perceptron, to improve the accuracy of breast cancer prediction. The methodology used in the paper involves selecting the best five machine learning algorithms, training them on the dataset, and then combining their predictions using logistic regression. The proposed system demonstrates improved performance compared to using individual machine learning algorithms.

According to [12] stacking as an ensemble learning strategy has universal properties since it can encompass and combine other regularly used ensemble learning approaches. The aforementioned ensemble learning systems, such as Voting, Selection by Cross-validation (X-Val), Grading, and Bagging, can be theoretically translated onto the stacking framework by introducing unique meta-classifiers. It can act as a unifying structure, allowing and extending other ensemble approaches. It functions as a meta-ensemble approach, capable of incorporating a wide range of ensemble strategies. Based on the knowledge gained from this effort, we chose to create our frameworks using the stacking technique.

Summarily, all the systems reviewed have provided a framework that is tailored towards a dataset and solves a task and that is solely because of the nature of the dataset used and the hyper parameter tuning. Our proposed system differs in that we aim to provide a framework that will take away the hassle of hyper parameter tuning, that will reduce the number of lines of codes and is extensible to accommodate more machine learning algorithms. We can use the framework to experiment with different dataset and different classification tasks.

2. METHODS

This section discusses the research framework, algorithm used in the development of the system and the steps for users of the system. The research framework for this research is given Figure 1.

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

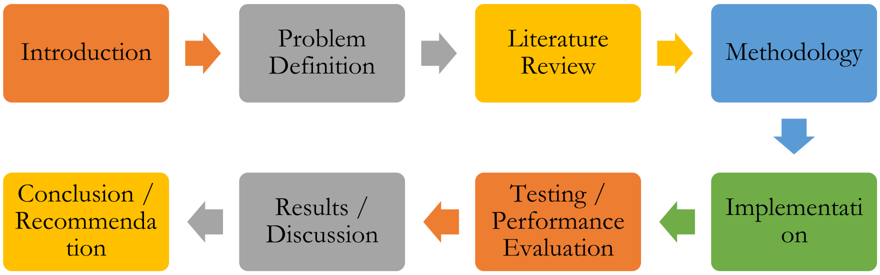

Figure 1. The Research Framework for EnsembleForge

Based on Figure 1 can be explained as follows.

1) The introduction highlights the significance of ensemble learning in machine learning, emphasizes the importance of integrating ensemble techniques in classification tasks. The EnsembleForge framework was introduced as a tool designed to simplify the training and deployment of stacked ensemble models, aiming to streamline the process for users. The objectives of the research were outlined, emphasizing the need for a comprehensive framework like EnsembleForge to enhance model performance and adaptability across various datasets and classification tasks.
2) During the literature review, the paradigm shift towards ensemble methods in machine learning was discussed, focusing on the benefits of combining diverse models to improve predictive accuracy. Existing research on ensemble learning, classification techniques, and model optimization challenges was reviewed, underscoring the necessity for advanced frameworks like EnsembleForge to overcome these obstacles and enhance model performance. The exploration of ensemble learning applications in different domains highlighted the potential impact of advanced ensemble frameworks in addressing complex classification problems.
3) The methodology: Here, the development algorithm and the algorithmic procedures for utilizing the EnsembleForge framework were described, including steps such as initialization, dataset preprocessing, model selection, hyperparameter tuning, and model evaluation. By following a systematic approach, users can effectively train and deploy stacked ensemble models using EnsembleForge, leveraging popular tools and libraries like Scikit-learn for efficient model development.
4) The implementation: Here, the actual development of the code was done, the Ensemble forge is primarily based on the scikit learn library. The programming language used for the development is python because of his ease of use and flexibility. Here, the object-oriented approach was employed to ensure the code base is modular and easy to maintain.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

5) Testing/performance Evaluation: Various components of the framework were tested before the overall testing of the entire system. During the testing, bugs and issues found during the implementation were analyzed and various sections of the algorithm was updated to reflect the updates. Various datasets where used and we employed RandaomSearchCv and GridSearchCv to cater for performance variations in the hyperparameter tuning when different datasets where used.
6) The results and discussion present the outcomes of testing the EnsembleForge framework with publicly available datasets, emphasizing its ease of use, adaptability, and performance compared to traditional methods. The framework's capabilities in optimizing model parameters and enhancing model accuracy were analyzed, demonstrating its potential to streamline the classification process and improve predictive outcomes.
7) Conclusion: the key findings and contributions of the EnsembleForge research work were summarized, highlighting the adaptability and innovation demonstrated by the framework in simplifying classification tasks. Future directions for extending EnsembleForge to support regression tasks and incorporate NLP capabilities were discussed, showcasing the framework's potential for further advancements in ensemble learning.
8) The recommendations: It provides suggestions for further research and enhancements to the EnsembleForge framework, encouraging researchers and practitioners to explore the capabilities of ensemble techniques for classification tasks. Guidance on leveraging EnsembleForge for experimentation with different datasets and machine learning algorithms was offered, promoting the adoption of advanced ensemble frameworks for improved model performance and efficiency.

The methodology for the Stacking Framework for Ensemble Learning is devised to offer a flexible and user-friendly approach, enabling users to seamlessly incorporate multiple machine learning models and hyper models.
The steps are highlighted below:
1) Import necessary libraries (e.g., Scikit-learn, EnsembleForge).
2) Initialize EnsembleForge with chosen models and preprocessing steps.
3) Preprocess the dataset using standard procedures.
4) Choose base models and meta models based on the problem.
5) Use hyperparameter tuning strategies for optimal configurations.
6) Train base models individually and evaluate performance.
7) Apply stacking ensemble to combine models.
8) Fine-tune meta model hyperparameters.
9) Train the stacking ensemble model.
10) Evaluate overall performance on the test set.
11) Visualize and analyze results.
12) Extend for other tasks by adapting models and integrating new ones.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The EnsembleForge class encapsulates the entire ensemble learning process, which incorporates popular algorithms like Random Forest, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN). The core components of the framework include:

1) EnsembleForge Class: This class is responsible for the entire ensemble learning process.
2) Base Models: A dictionary of base classifiers and their optional hyperparameters is provided by the user, allowing flexibility in choosing diverse algorithms for the ensemble.
3) Preprocessing Steps: Optional preprocessing steps, like scaling using StandardScaler, can be applied to the data before training the base models.
4) Hyperparameter Tuning: The framework incorporates hyperparameter tuning using RandomizedSearchCV during the training of base models.
5) Feature Importance Analysis: The framework checks if each base model supports feature importance analysis. If supported, it logs feature importances, providing insights into the contribution of each feature to model predictions.
6) Logging and Error Handling: The framework employs logging for error handling during the training of base models, ensuring robustness.

2.1 Algorithm for Ensembleforge

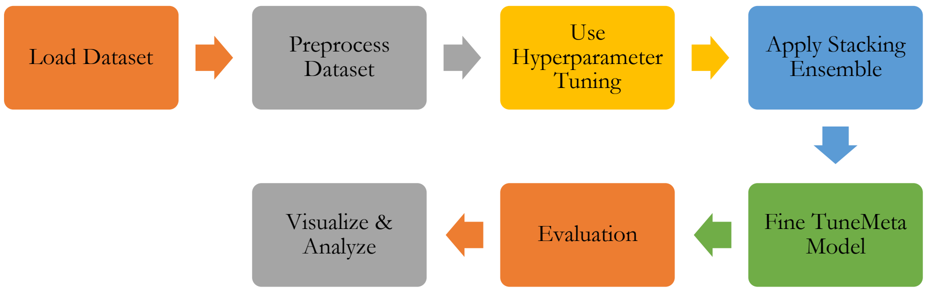

Figure 2. Process for EnsembleForge

2.2 Stacking Model Creation

The stacking ensemble model is constructed by combining the trained base Classifiers with the specified meta model. The interaction between individual base models and the meta model is crucial to the ensemble's performance. The image below shows the architecture of the stacked ensembled framework.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

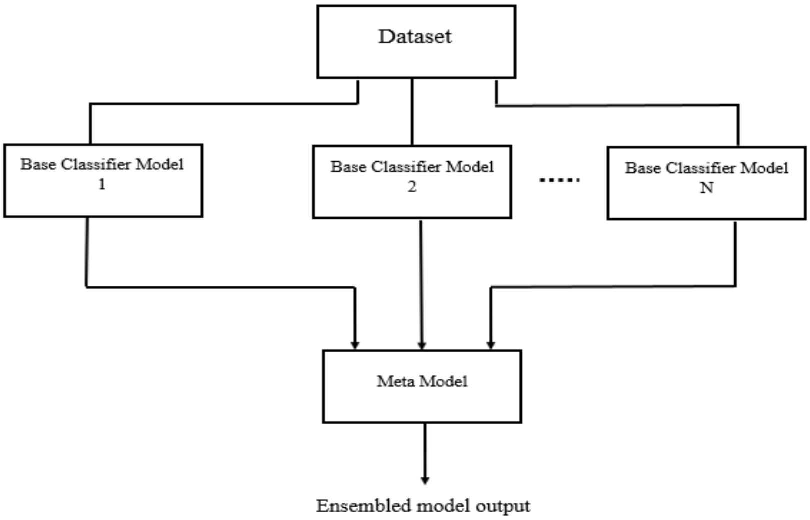

Figure 3. The Ensemble framework Stacking Process

3. RESULTS AND DISCUSSION

The proposed framework **(EnsembleForge)** is built on the Scikit-learn Library by [11]. Scikit-learn, sometimes known as sklearn, is a popular and versatile Python machine learning framework. It provides a comprehensive collection of tools for building and evaluating machine learning models, including classification, regression, clustering, and dimensionality reduction. The implementation is conceived with the primary objective of furnishing a convenient avenue for constructing and assessing classification models. This is achieved by streamlining the intricacies associated with hyperparameter tuning, dataset preprocessing, and performance evaluation. The framework aims to offer users these essential functionalities with minimal code, simplifying the process of building and evaluating classification models.

3.1 Initialization

The EnsembleForge class initializes the entire ensemble learning process, allowing users to provide essential parameters:

1) **Base Classifiers:** A dictionary of base classifiers and their optional hyperparameters.
2) **Hypermodel Name:** The name of the hypermodel (meta-model) responsible for combining base models in the ensemble.
3) **Optional Preprocessing Steps:** Users can specify preprocessing steps, such as scaling using StandardScaler.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Table 1 gives a summary of the initialization parameters, their significance and potential values to the framework.

Table 1. Initialization Parameters to the framework

Parameters	**Parameter name**	**Significance**	**Potential value**
Base Classifiers and Hyperparameters:	base_classifiers	Dictates the choice of base classifiers for the ensemble, allowing users to tailor the framework to their specific needs.	A dictionary where keys are classifier names, and values are tuples containing the classifier and optional hyperparameters.
Hypermodel Name	meta_model_name	Specifies the hypermodel (meta-model) responsible for combining base models in the ensemble	String representing the hypermodel name (e.g., 'LogisticRegression', 'RandomForestClassifier').
Optional Preprocessing Steps:	preprocessing_steps	Allows users to specify optional preprocessing steps applied to the data before training the base models, enhancing adaptability to various datasets.	List of tuples where each tuple consists of a string identifier for the preprocessing step and the corresponding scikit-learn transformer or estimator.
Meta-model Hyperparameters	meta_hyperparameters	Governs the hyperparameters of the meta-model, influencing the	A dictionary representing hyperparameters for the specified meta-model. Users can provide values such as learning rates,

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Parameters	Parameter name	Significance	Potential value
		ensemble's behavior	regularization parameters, or tree depths.

3.2 Base Model Training

For each base classifier specified by the user, the Stacking Framework follows these steps:

1) A pipeline is created, incorporating optional preprocessing steps for data preparation.
2) Hyperparameter tuning is performed using RandomizedSearchCV or GrdidSearchCV, optimizing the model's configuration.
3) The best model is then trained on the entire training set.

3.3 Base Classifiers Default Hyperparameters:

If users do not provide specific hyperparameters for base classifiers, default configurations are applied based on the individual model's characteristics. Table 2 shows default hyper parameters.

Table 2. Default hyperparameters for the Base Classifiers

Base Classifier	Default configurations
Random Forest(RF)	{'n_estimators': 100, 'max_depth': None}
Gradient Boosting(GB)	{'n_estimators': 100, 'learning_rate': 0.1}
Support Vector Machines (SVM)	{'C': 1.0, 'kernel': 'rbf'}
Logistic Regression(LR)	{'C': 1.0}
K- Nearest neighbor (KNN)	{'n_neighbors': 5}
NaiveBayes	{}

3.4. Meta-model:

If users do not provide specific hyperparameters for the meta-model, default configurations are applied based on the chosen meta-model. For example, Logistic Regression might use {'C': 1.0, 'penalty': 'l2'} as default hyperparameters.

3.5. Ensemble Model Training

The stacking ensemble model is trained on the provided training set, incorporating the knowledge acquired from the base models as shown in Figure 1. The Table 3 below shows the meta-model used in the ensemble stage. These default parameters are used when the user does not provide hyperparameters.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Table 3. Meta model Default Hyperparameters

Meta-Model	Meta-Model Name	Default Hyperparameters
Logistic Regression(LR)	'LogisticRegression'	{'C': 1.0, 'penalty': 'l2'}
XGBoost Classifier	'XGBClassifier'	{'objective': 'binary:logistic', 'use_label_encoder': False, 'eval_metric': 'logloss'}
LightGBM Classifier	'LGBMClassifier'	{'objective': 'binary', 'metric': 'binary_logloss', 'boosting_type': 'gbdt', 'num_leaves': 31, 'learning_rate': 0.05, 'feature_fraction': 0.9}

The default parameters for the XBGClassifier and the LGBMClassifer depends on the version of the library available in the user's environment. More meta classifiers can be added to the framework as required.

3.6. Evaluation

The ensemble model's performance is evaluated using standard metrics such as accuracy, confusion matrix, and classification report. The rationale behind the chosen metrics is provided. The image below shows the output from the proposed system for two base models of K-nearest neighbor (KNN) and support vector machines SVM and logistic regression as the meta model on the breast cancer dataset.

Table 4. Performance Report of the Proposed System on the Breast Cancer Dataset by [16]

Model	Accuracy	Precision	Recall	F1-score
KNN	95%	95%	95%	95%
SVM	98%	98%	98%	98%
Stacking Ensemble	96%	96%	96%	96%

The performance metrics available to evaluate the performance of the framework are explained below:

1) **Accuracy**: Gives an overall measure of the correct classifications. Suitable when classes are balanced; however, be cautious in imbalanced datasets.
2) **Precision, Recall, and F1 Score**: Provide insights into the performance of the model for each class. Especially crucial when dealing with imbalanced datasets or when the cost of false positives/negatives varies.

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

3) **Confusion Matrix**: Gives a detailed breakdown of true positives, true negatives, false positives, and false negatives. Useful for understanding the types and quantities of errors made by the model.

3.7. Visualization

The framework provides a method to train and evaluate the performances of each model and generate visual representations of the ensemble model's performance. The figure below shows the visual representation of the models' performances. This is necessary to investigate the performance of each model on the data and the overall performance of the ensemble model. This can help inform about the model to tune for better accuracy.

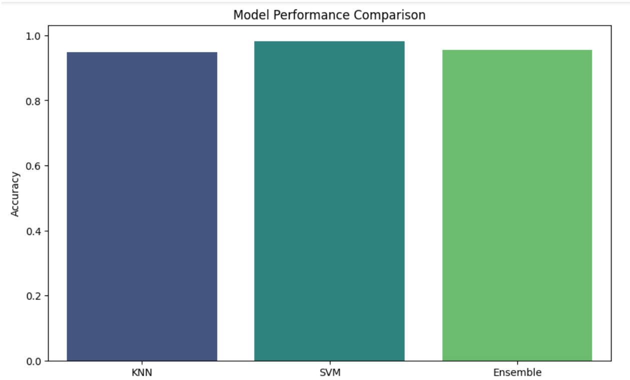

Figure 4. Frameworks Model Performance Comparison Visualization

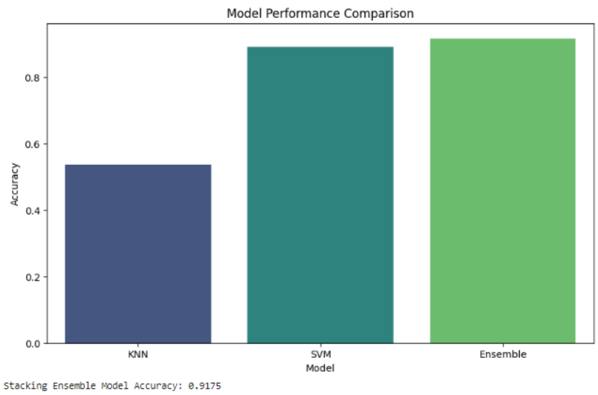

Figure 5. Model Visualization on the Phone Price Classification Dataset by [6]

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

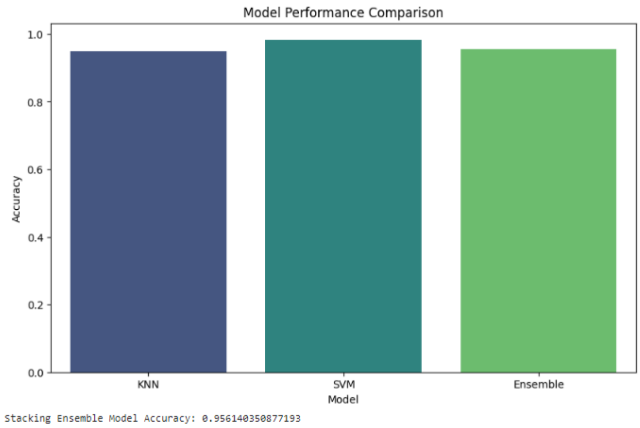

Figure 6. Model Performance Visualization on Breast Cancer Dataset by [14]

Table 4. Evaluation Results of the Framework.

Dataset	Base Models	Meta Model	Precision (%)	Recall (%)	F1 score (%)	Ensemble Accuracy	Classification Type
Breast cancer	RF, KNN, SVM	LR	96	95	96	95	Binary
Breast cancer	KNN, NaiveBayes, LR	RF	97	97	97	97	Binary
Breast cancer	KNN, NaiveBayes, LR	XGBClassifier	97	97	97	97	Binary
Breast cancer	KNN, SVM	LR	95	95	95	95	Binary
Wine	KNN, SVM	LR	100	100	100	100	Multiclass
Wine	RF, KNN	LR	97	97	97	97	Multiclass
Phone Price	KNN, NaïveBayes	RF	82	81	81	81	Multiclass
Phone Price	KNN, NaiveBayes, LR	RF	96	96	95	96	Multiclass
Phone Price	KNN, SVM	LR	92	91	91	91	Multiclass
Digits	KNN, SVM, RF	LR	98	98	97	98	Multiclass

Table 4 shows the EnsembleForge framework's thorough classification report, which highlights its performance across several base models.

The EnsembleForge framework has been meticulously developed as a user-friendly and versatile tool for ensemble learning, allowing researchers, machine

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

learning enthusiasts, and students to explore various ensemble models effortlessly. Orchestrating the ensemble learning process, the EnsembleForge class supports diverse base classifiers and meta-models, incorporating optional preparation measures like scaling and hyperparameter tuning to enhance adaptability to different datasets. Feature importance analysis and error recording during base model training ensure robustness and flexibility. The framework excels in optimizing model parameters through approaches like RandomizedSearchCV and GridSearchCV. While adept at handling classification problems, it recognizes the need for personalized preprocessing steps in certain datasets, addressing issues such as class imbalances, null values, and text label outputs. Ongoing efforts involve extending EnsembleForge to support regression tasks and incorporating NLP capabilities, aiming to provide a unified platform for users to explore ensemble learning across classification, regression, and NLP domains.

4. CONCLUSION

In conclusion, EnsembleForge represents a groundbreaking advancement in machine learning, offering researchers and practitioners a versatile and efficient framework for tackling classification tasks. With its adaptability and ease of use, EnsembleForge streamlines the process of model development and experimentation, empowering users to achieve greater accuracy without extensive coding. As we push the boundaries of machine learning capabilities, EnsembleForge stands as a testament to innovation and creativity in the field.

REFERENCES

[1] K. M. R. Alam, N. Siddique, and H. Adeli, "A dynamic ensemble learning algorithm for neural networks," *Neural Computing and Applications*, vol. 32, pp. 8675-8690, 2020.
[2] D. B. Araya, K. Grolinger, H. F. ElYamany, M. A. Capretz, and G. Bitsuamlak, "An ensemble learning framework for anomaly detection in building energy consumption," *Energy and Buildings*, vol. 144, pp. 191-206, 2017.
[3] A. Chatzimparmpas, R. M. Martins, K. Kucher, and A. Kerren, "StackGenVis: Alignment of data, algorithms, and models for stacking ensemble learning using performance metrics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1547-1557, 2020.
[4] V. Chaurasia and S. Pal, "An ensemble framework-stacking and feature selection technique for detection of breast cancer," *International Journal of Medical Engineering and Informatics*, vol. 14, no. 3, pp. 240-251, 2022.
[5] Y. Chen, Y. Zhou, M. Park, S. Tran, S. Hadley, and Q. Bai, "A novel adaptive ensemble learning framework for automated Beggiatoa Spp. coverage estimation," *Expert Systems with Applications*, vol. 237, p. 121416, 2023.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

[6] J. Divakarr, "Phone Classification Dataset," Kaggle, Retrieved January 9, 2024, from https://www.kaggle.com/datasets/jacksondivakarr/phone-classification-dataset.

[7] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *Journal of King Saud University-Computer and Information Sciences*, 2023.

[8] A. Özdemir, U. Yavuz, and F. A. Dael, "Performance evaluation of different classification techniques using different datasets," *International Journal of Electrical & Computer Engineering*, vol. 9, no. 5, 2019.

[9] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *the Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[10] P. Pintelas and I. E. Livieris, "Special issue on ensemble learning and applications," *Algorithms*, vol. 13, no. 6, p. 140, 2020.

[11] L. Ren, H. Zhang, A. S. Seklouli, T. Wang, and A. Bouras, "Stacking-based multi-objective ensemble framework for prediction of hypertension," *Expert Systems with Applications*, vol. 215, p. 119351, 2023.

[12] A. K. Seewald, "Towards a theoretical framework for ensemble classification," *in IJCAI*, vol. 3, pp. 1443-1444, 2003.

[13] N. Singh and P. Singh, "Stacking-based multi-objective evolutionary ensemble framework for prediction of diabetes mellitus," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 1-22, 2020.

[14] W. Wolberg, "Breast Cancer Wisconsin (Original) [Data file]," UCI Machine Learning Repository, https://doi.org/10.24432/C5HP4Z, 1992.

[15] Y. Zhang, J. Liu, and W. Shen, "A review of ensemble learning algorithms used in remote sensing applications," *Applied Sciences*, vol. 12, no. 17, p. 8654, 2022.