Food Classification Model

Suhana Islam
Department of Electrical and Computer Engineering
North South University
Dhaka,Bangladesh
suhana.islam@northsouth.edu

Nafis Anzum
Department of Electrical and Computer Engineering
North South University
Dhaka,Bangladesh
nafis.anzum@northsouth.edu

Tasfia Anjum Zuairia
Department of Electrical and Computer Engineering
North South University
Dhaka,Bangladesh
anjum.zuairia@northsouth.edu

Ekfat Jahan Ashrafy
Department of Electrical and Computer Engineering
North South University
Dhaka,Bangladesh
ekfat.nowshin@northsouth.edu

***Abstract*—This study focuses on developing a classification model using Logistic Regression and Decision Tree algorithms, which was further improved through the use of ensemble learning techniques to enhance both accuracy and robustness in predicting whether the food is healthy or unhealthy.A sample dataset containing details about various food items and their nutritional contents were utilized. We implemented several ensemble methods – including Random Forest, AdaBoost, XGBoost and Soft Voting – and performed hypertuning on each. These models were then evaluated against the baseline models from Phase 1 . The results showed improved accuracy and generalization with XGBoost delivering the best performance.**

Keywords—Logistic Regression , Decision Tree,Ensemble Techniques, Random Forest, AdaBoost, XGBoost, Soft Voting, Evaluation Metrics, Accuracy, Precision, F1-Score, Recall, Heatmaps, Box Plots, Hyperparameter Tuning

I. INTRODUCTION (*HEADING 1*)

In recent years, public health awarness has grown significantly , particularly concerning food consumption and its impact on long-term well-being and recent research emphasizes the need to understand the importance of nurtitional composition of food and the role it plays in our lifestyle related diseases such as diabetes, obesity and cardiovascular conditions. Classifying food into "healthy" and "unhealthy" categories is essential for promoting balanced diets. This classification not only helps the individual consumers in maintaining a healthy lifestyle but also the researchers , policymakers and consumers in making informed dietary choices.

In Phase 1 of our project, we developed two baseline models – Logistic Regression and Decision Tree Classifier – to categorize foods based on their nutritional content. Intially we preprocessed the dataset by dropping and imputing. After training and hyperparameter tuning the models, we compared the pre and post tuning results for each of the model. While both models performed reasonably well, the Decision Tree exhibited slight overfitting.

To address this limitation and improve the performance , Phase 2 focused on implementing ensemble learning techniques to enhance accuracy, robustness and generalizability.

Our main contributions in project include:

- **Problem Definition :** The goal is to build a reliable classifier that can predict the health status of food items based on the nutritional content
- **Data Preprocessing :** With the aid of dropping and imputation, unnecessary columns were eliminated
- **Exploratory Data Anaylsis :** Correlation Heatmap, Boxplot, Scatter Plot, Histogram was used to visualize the preprocessed data
- **Model Training and Evaluation:** Splitting ratio was considered 80%(training) and 20%(testing) in Phase 1 and 70%(training), 15%(validation), 15%(testing)
- **Hyperparameter Tuning :** GridSearch was used to find the best parameters for each model
- **Application of Multiple Ensemble Models:** Random Forest, AdaBoost, XGBoost was implemented to ensemble the Soft Voting Classifier Model
- **Comparative Analysis of Models :** The performance of the Phase 1 models were comapred with the performance of Phase 2 models.
- **Visualization and Performance Analysis:** BoxPlot, Heatmap and Comparison tables were used to assess the performance

II. LITERATURE REVIEW

Machine learning is a subfield of artificial intelligence that can learn patterns from data and make predictions witout being explicitly programmed.Machine learning algorithms are widely used in computer science domains. These problems are almost used for classification and regression problems in every aspect of life. Machine learning models generalize data to predict outcomes on unseen data. Depending on the problem, machine learning technqiues can be divided into three categories: supervised,unsupervised an reinforcement.

Supervised learning, which works primarily on labelled data, can produce a prediction more easily compared to

unsupervised learning. On the other hand, unsupervised learning which works on the basis of unlabelled data is mostly used to discover hidden patterns, groupings and structures.

Machine learning models can vary in complexity, interpredability, and performance. Choosing the right model depends of the given problem currently being dealt with. Our goal is to balance between interpredability and accuracy, and coputational efficiency. The supervised learning is mainly categorized into two types- Classification and Regression problems. Logistic regression is a classification linear model which is used to solve binary classification problems. It predicts the output of a given input using the logistic(sigmoid function). The output lies between 0 and 1. Decision Tree is a model that can handle both numerical and categorical data. It works by splitting the dataset based on the feature values and forming a tree like structure.

Ensemble learning is the process of combining predictions from multiple machine learning models. Ensemble Learning has gained widespread popularity in recent years for its ability to improve prediction accuracy by combining the strengths of multiple base models. It gives better productive performance than any individual model can give alone. Bagging methods like Random Forest reduce variance by training multiple estimators on different subsets of data, while Boosting methods such as XGBoost focus on reducing bias by sequentially correcting errors of prior models. Voting and stacking strategies combine predictions from multiple models to further improve generalization. These techniques hae been applied in various sectors including healthcare, finance and natural language processing.

III. Methodology

This section outlines the methodology including the additional preprocessing, application of ensemble models, training process and comparison of models between both phases.

A. Dataset Analysis

The dataset used in the study contains detailed data about food production in Brazil , next to attributes like pesticides usage, harvest area, food categories, quantity applied, toxicity levels, and various numeric measures related to food production. The dataset seems to be compiled from multiple official and agricultural sources, providing a strong foundation for pattern analysis whether a food is healthy or unhealthy. The dataset consists of 439,299 food items and 51 features,including nutrient composition and environmental indicators. The thoroughness of the dataset allows the exploration of such a target which are essential modelling and prediction.

The dataset was imported using panda, using head{} for an initial examination and info( ) functions to comprehend the structure and data types and generation for the numerical distritbutions using describe( ).

B. Exploratory Data Analysis (EDA)

Exploratory Data Analysis summarize the main characteristics of a dataset using visual methods. The goal of EDA is to understand the data structure and contents, detect outliers, identifying missing values. It helps us to discover relationship between features.

Histogram was used for visualizing feature distribution.

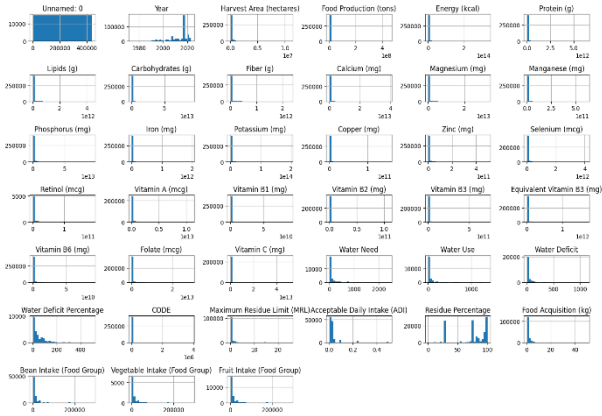

Fig: Histogram for Numerical Distribution

Correlation heatmap for numerical features to identify the highly correlated features.

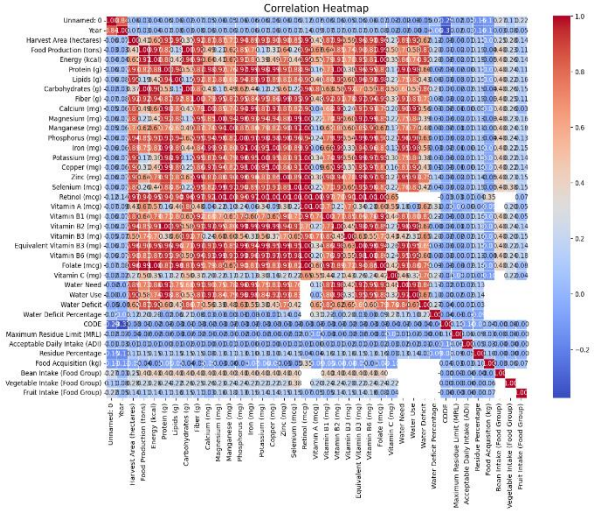

Fig: Correlation Heatmap

To detect the potential outliers, interquartile range was used and with the help of boxplot, the data ditribution before and after outlier removal was represented.

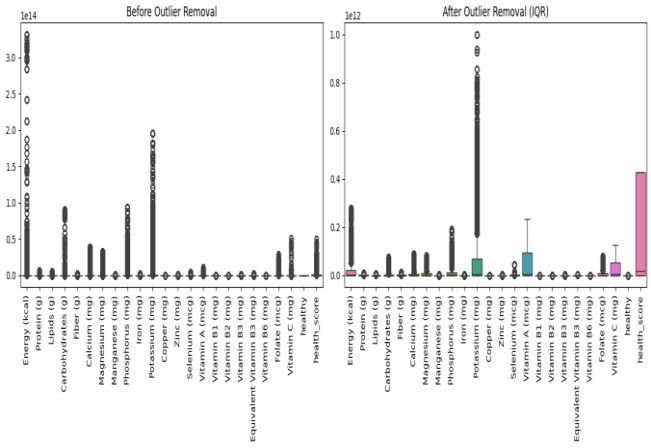

Fig : BoxPlot Distribution Before and After Outlier Removal

The missing value percentage was checked to drop in preprocessing step.

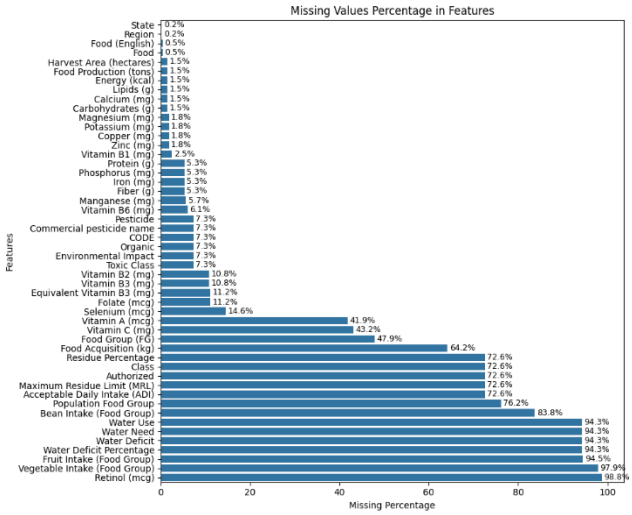

Fig : Missing Value Percentage

The target class distribution was checked to verify whether it was balanced.

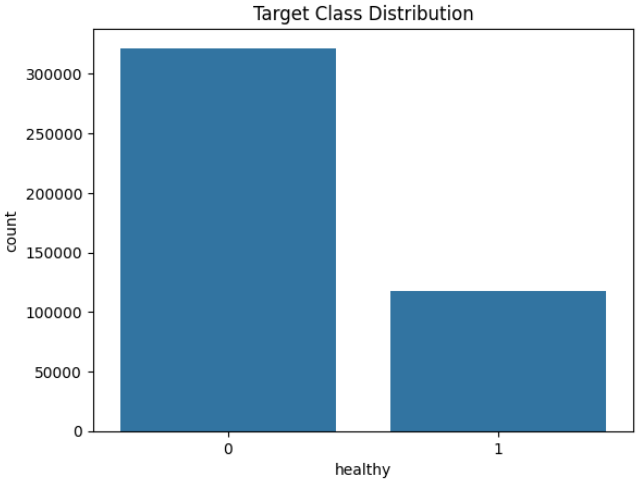

Fig: Class Balance

C. *Target Label Creation*

Due to lack of target variable,a binary classification target column “healthy” and “health_score” was created based on Energy, Protein, Fiber, Vitamin A, Vitamin C, Iron, Calcium, Lipids, and Carbohydrates. If the food is above the medium score, then it is classified as healthy=1 otherwise healthy=0. After the target variable creation, correlation with target column to check if any extra highly correlated column was necessary to drop.

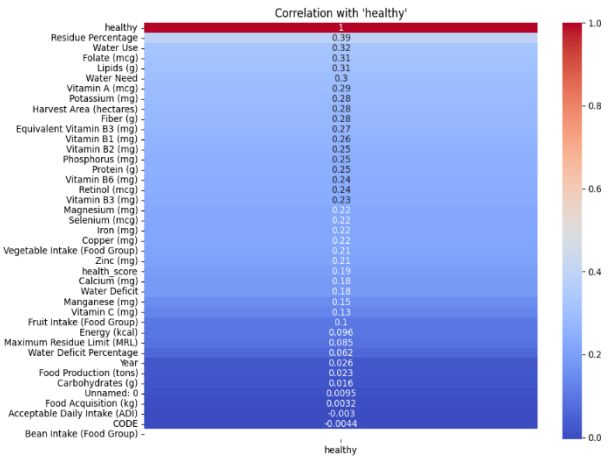

Fig: Correlation with Target Variable

D. *Data Preprocessing*

Data Preprocessing is a very crucial step for training an accurate model. It helps to eliminate any unwanted data from the dataset to avoid redundancy.

Handling Missing Values: For columns with more than 50% missing values were dropped and also rows missing target varibale were also dropped.
Imputation: We filled numerical columns with mean and the categorical columns with mode. Both were performed using Simple Imputer from sklearn.impute.
Outlier Removal: The interquartile range was used to detect the outliers and then outliers were removed.

IQR= Q3-Q1

Encoding: One-hot Encoding was used to convert categorical columns into numercial columns.

E. *Splitting the Data*

The dataset was splitted into 70% train set, 30% test set to avoid data leakage. The target variable “healthy”,“health_score” were dropped. Furthermore, the 30% test set were splitted into 15% validation and 15% test set. The SMOTE was applied only on training set.

F. *Model Training and Evaluation*

Three different models were trained-Random Forest,AdaBoost,XGBoost. The output of these models were fed into Soft Voting Classifier as inputs. The three models were tested on the test set and their corresponding outputs were observed individually.

IV. Experiment

All experiments were conducted using **Python 3** within a **Jupyter Notebook** environment, leveraging powerful machine learning libraries such as **scikit-learn**, **XGBoost**, and **matplotlib/seaborn** for modeling, tuning, and visualization. The dataset used was split into **80% training** and **20% testing** to ensure robust evaluation while maintaining sufficient data for model learning. Prior to training, **standardization** was applied to all numerical features using StandardScaler to ensure uniform feature scales, which is particularly important for gradient-based

algorithms and distance-based models to perform efficiently and converge effectively.

To evaluate model performance under consistent conditions, we maintained the same data split for all models across experiments. This consistency allowed fair and reliable comparisons between different ensemble methods such as **Random Forest**, **XGBoost**, and **Voting Classifier**, as well as the Phase 1 baseline models. For each ensemble model, we conducted **hyperparameter tuning** using **GridSearchCV** with **5-fold cross-validation**. This technique systematically searches through predefined hyperparameter combinations to identify the configuration that yields the best cross-validated performance.

The evaluation metrics used include:

- **Accuracy**: The proportion of correctly classified samples among all predictions.
- **Precision**: The ability of the model to correctly identify foods labeled as healthy out of all those predicted as healthy.
- **Recall**: The model's ability to correctly detect all actual healthy foods in the dataset.
- **F1 Score**: The harmonic mean of precision and recall, especially useful for imbalanced datasets.
- **Confusion Matrix**: A matrix that provides a detailed breakdown of classification results across four categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

Each metric was chosen to offer a comprehensive view of performance, particularly under imbalanced class scenarios where a single metric like accuracy might not fully capture model behavior.

GridSearchCV was employed to tune critical hyperparameters for the ensemble models. For example:

- In **Random Forest**, key hyperparameters included n_estimators, max_depth, min_samples_split, and criterion.
- In **XGBoost**, parameters such as learning_rate, max_depth, n_estimators, subsample, and colsample_bytree were optimized.

Although **Grid Search** is computationally intensive—since it exhaustively evaluates all possible combinations in the parameter grid—it was chosen for its simplicity and thoroughness. The tuning process was essential to improve model generalization and reduce overfitting, ensuring that the trained models perform effectively on unseen data rather than just the training set.

Furthermore, **Voting Classifier**, which combines predictions from multiple models, was tuned by adjusting the weights assigned to each base model to balance their influence. This allowed us to exploit the strengths of each model while minimizing their individual weaknesses.

In conclusion, the experimental design was aimed at maximizing model performance through careful preprocessing, consistent evaluation, and rigorous tuning. The combination of multiple evaluation metrics and hyperparameter optimization techniques led to a well-rounded assessment and robust implementation of ensemble learning techniques.

The results for pre and post hypertuning are given below:

Results for Pre-Tuning:

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Random Forest	1	1	1	1
AdaBoost	0.99	0.99	0.98	0.99
XGBoost	1	1	1	1

Results for Post-Tuning:

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Random Forest	0.99985	1	0.99943	0.999972
AdaBoost	0.999894	0.999943	0.999660	0.999802
XGBoost	0.999879	0.999830	0.999717	0.999773
Ensemble Model(Soft Voting)	0.999970	1	0.999887	0.999943

After performing hyperparameter tuning, the results of the models were represented using confusion metrics as given below:

Random Forest:

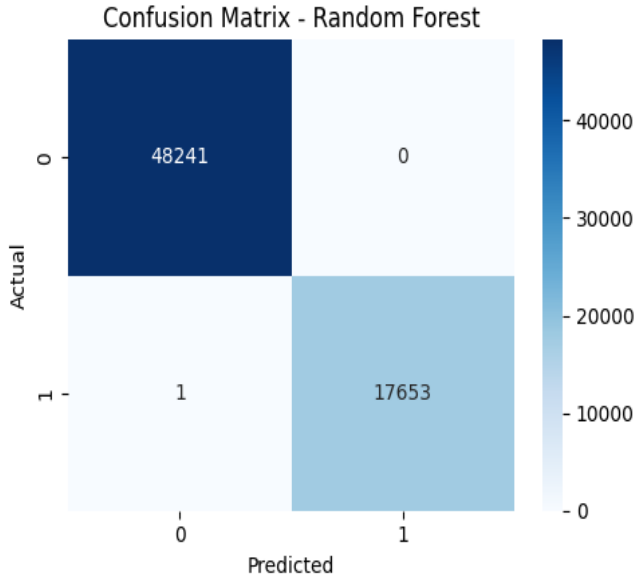

Fig: Confusion Matrix for Random Forest

AdaBoost:

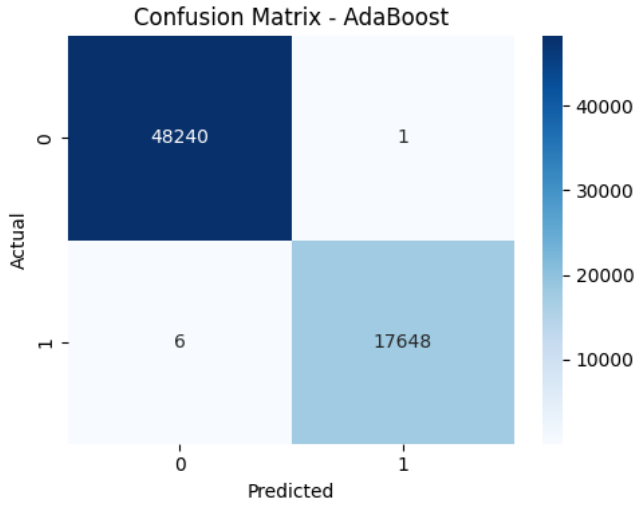

Fig: Confusion Matrix for AdaBoost

XGBoost:

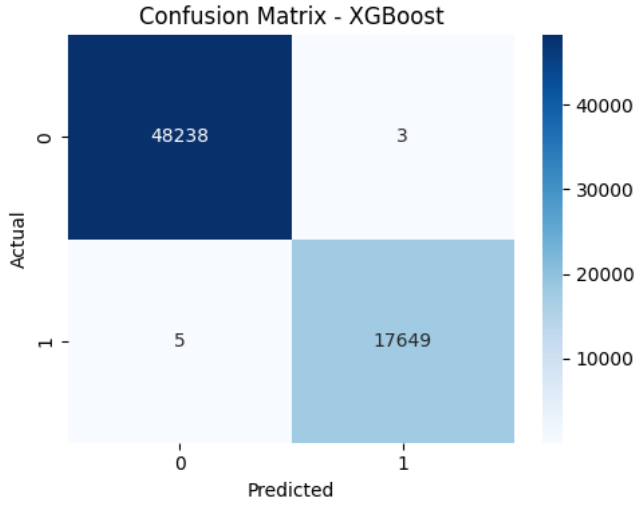

Fig: Confusion Matrix for XGBoost

Soft Voting:

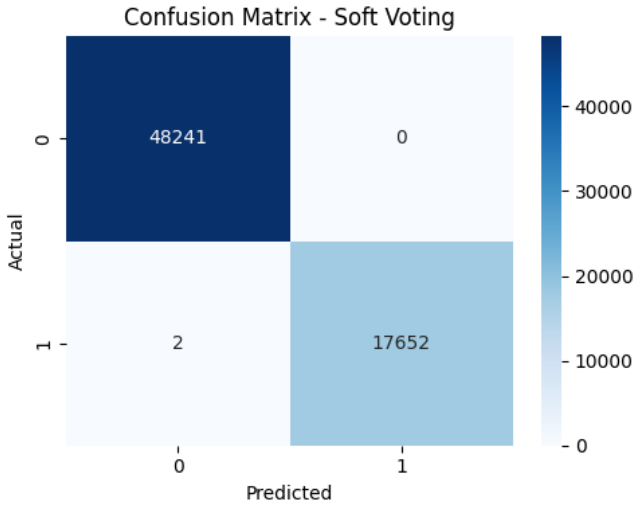

Fig: Confusion Matrix for Soft Voting

V. Discussion

Throughout the project, we encountered several strengths and weaknesses that shaped the outcomes and guided our decision-making process. One of the most notable strengths was our ability to significantly improve model accuracy and generalization by implementing advanced ensemble learning techniques such as Random Forest, XGBoost, and Voting Classifier. These methods provided a substantial boost in classification performance compared to the baseline models developed in Phase 1 (Logistic Regression and Decision Tree). The ensemble models not only achieved better predictive accuracy but also demonstrated improved robustness across various test scenarios. Specifically, they helped mitigate overfitting issues that were previously observed in the Decision Tree model, leading to more stable and reliable predictions.

To ensure a comprehensive assessment of model performance, we employed a diverse set of evaluation metrics, including accuracy, precision, recall, F1-score, and confusion matrices. These metrics allowed us to capture different aspects of model behavior, particularly in the context of imbalanced data where accuracy alone may be misleading. By analyzing these metrics together, we gained a deeper and more nuanced understanding of how each model performed in distinguishing between healthy and unhealthy food categories.

Another key strength was the application of hyperparameter tuning using Grid Search. This technique enabled us to systematically explore a wide range of hyperparameter combinations for each ensemble model, ultimately leading to more optimized configurations. As a result, we were able to enhance model performance while also reducing unnecessary computational costs. Grid Search ensured that each model was fine-tuned for both efficiency and effectiveness.

In addition, visualization techniques played an essential role in our project. Tools such as boxplots, heatmaps, and comparative performance tables allowed us to visually interpret model behavior and evaluate performance trade-offs across different algorithms. These visual aids not only made it easier to communicate our findings but also helped in identifying patterns and anomalies that might not have been obvious through numerical metrics alone.

Despite these strengths, there were also some challenges and trade-offs. Ensemble methods, while powerful, often increase model complexity and reduce interpretability, making it more difficult to explain the reasoning behind specific predictions. Moreover, the computational demands for training and tuning complex models like XGBoost can be substantial, especially when dealing with large feature spaces or running extensive Grid Search procedures.

Overall, the strengths of ensemble learning, combined with robust evaluation strategies and thoughtful visualization, enabled us to significantly enhance the performance of our food classification system. The challenges we encountered provided valuable lessons in balancing accuracy, efficiency, and interpretability—insights that will be crucial for future research and real-world applications in the health and nutrition domain.

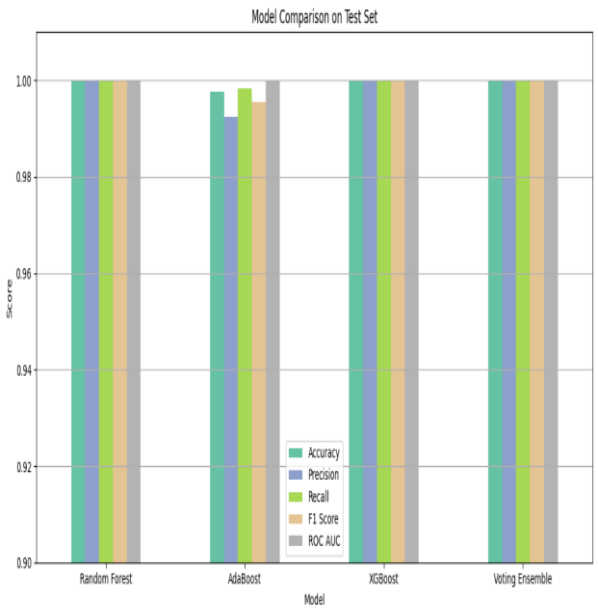

Fig: Evaluation Metrics of Ensemble Models

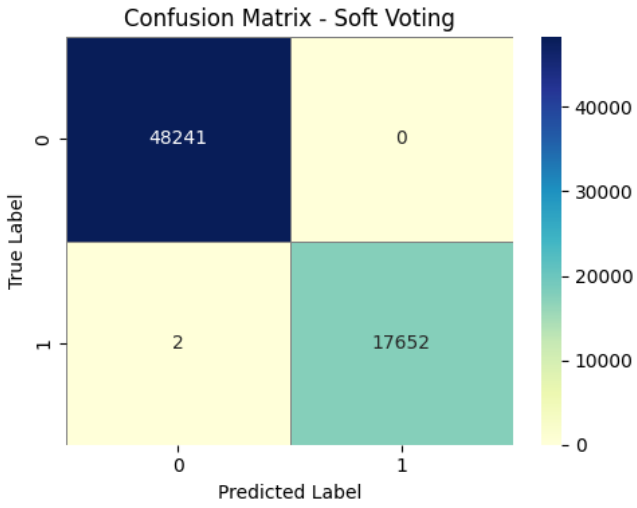

FIG: FINAL RESULT OF SOFT VOTING

Comparison with Phase 1 Models:

In Phase 1 , the models is simple with low computational complexity .Its easier to interpret but prone to overfitting and generalization. On the other hand Phase 2 utilize complex ensemble models and hyperparameter tuning (GridSearch) for better performance.

It has a higher computational complexity than Phase 1.the Phase 2 models is more scalable and better generalised to unseen data.

The Phase 1 model is limited in accuracy but the Phase 2 model performs significantly specially better in precision, recall due to utilization of ensemble model.

VI.CONCLUSION

In this project, we explored the classification of food items into healthy and unhealthy categories using both baseline and ensemble machine learning models. In Phase 1, we implemented Logistic Regression and Decision Tree classifiers, which provided a foundational understanding of model performance based on nutritional data. However, to overcome limitations such as overfitting and limited generalization, Phase 2 introduced ensemble learning techniques—Random Forest, XGBoost, and a Voting Classifier—which significantly enhanced the robustness, accuracy, and reliability of predictions.

Several sectors still remains for future exploration. One potential

While our ensemble models significantly improved classification performance, several avenues remain for future exploration. One potential direction is the integration of deep learning models, such as Convolutional Neural Networks (CNNs), especially if image or ingredient-level data becomes available. Additionally, incorporating domain-

specific features like dietary guidelines or user demographics could enhance model context and personalization. Future work could also focus on explainable AI (XAI) techniques to make predictions more interpretable for healthcare professionals and consumers. Lastly, deploying the trained models in a real-world application, such as a mobile app for food scanning and health recommendations, could translate this research into impactful societal use.

For a more personzalided prediction,we can add domain specific features, we can deploy the model in real world like a mobile app or a website .

We explored different Machine learning models and used a powerful ensemble model techniques like Random Forest, Adaboost, XGBoost, and a Voting Classifier. Eventho our model performed well but there is still room to grow. WE can apply deep learning models to intreduce image data.

ACKNOWLEDGMENT *(Heading 5)*

WE WOULD LIKE TO EXPRESS OUR SINCERE GRATITUDE TO OUR SUPERVISOR, SILVIA AHMED, OF THE DEPARTMENT OF COMPUTER SCIENCE, FOR HIS INVALUABLE GUIDANCE AND UNWAVERING SUPPORT THROUGHOUT THIS PROJECT. ADDITIONALLY, WE EXTEND OUR DEEPEST APPRECIATION TO THE DEVELOPERS OF OLLAMA AND CHROMADB FOR PROVIDING THE ROBUST TOOLS THAT ENABLED THE SUCCESSFUL COMPLETION OF THIS ENDEAVOR. REFERENCES

[1] https://www.geeksforgeeks.org/handling-imbalanced-data-for-classification/

[2] https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/

[3] https://scikit-learn.org/stable/modules/impute.html

[4] https://www.datacamp.com/tutorial/one-hot-encoding-python-tutorial

[5] https://www.datacamp.com/tutorial/random-forests-classifier-python

[6] https://medium.com/@weidagang/essential-python-for-machine-learning-xgboost-4b662cf19fcd

[7] https://medium.com/@enozeren/building-the-adaboost-model-from-scratch-with-python-db3a8a 763484

[8] https://www.geeksforgeeks.org/box-plot-in-python-using-matplotlib/

[9] https://www.w3schools.com/python/matplotlib_histograms.asp

[10] https://www.kaggle.com/code/holfyuen/tutorial-scatter-plots-in-python

[11] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier

[12] .html https://www.analyticsvidhya.com/blog/2021/04/how-to-handle-missing-values-of-categorical-vari ables/

[13] https://medium.com/@paghadalsneh/handling-missing-numerical-data-techniques-and-impleme ntations-7371973c9039

[14] https://www.geeksforgeeks.org/drop-rows-from-pandas-dataframe-with-missing-values-or-nan-in-columns/ https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.

[15] html https://www.digitalocean.com/community/tutorials/grid-searching-using-python

[16] https://scikit-learn.org/stable/modules/cross_validation.html

[17] https://www.geeksforgeeks.org/detect-and-remove-the-outliers-using-python/

[18] https://www.datacamp.com/tutorial/understanding-logistic-regression-python

[19] https://www.datacamp.com/tutorial/decision-tree-classification-python

[20] https://www.programiz.com/python-programming/pandas/dataframe