**MODEL REFINEMENT**

**Machine Learning-Based Wheat Yield Prediction (የስንዴ ምርት ትንበያ) Using Environmental Factors**

**Group 15**

**Name:** Lielina Akele **Github Username:** @Lielina19

**Name:** Melat Kebede  **Github Username:** @MelatKebedeAbraham

**Model Refinement**

1. **Overview**

In this phase of our project, we concentrated on refining the machine learning model developed for predicting wheat yield in Ethiopia, a critical agricultural challenge in the region. Following the initial model exploration, we identified several performance gaps, such as low R² scores and signs of overfitting, which hindered the model’s ability to generalize effectively. Our primary objective during this refinement stage was to systematically address these shortcomings by optimizing the model architecture, improving data preprocessing, and enhancing predictive accuracy using environmental data like temperature, humidity, and precipitation. This effort aimed to create a more reliable tool for farmers and policymakers in Ethiopia to anticipate crop yields amidst changing climatic conditions.

1. **Model Evaluation**

We began by thoroughly evaluating the initial set of models we had developed, which included Linear Regression, LightGBM, Support Vector Regression (SVR), Convolutional Neural Network (CNN), Random Forest, XGBoost, and Decision Tree. The results were disappointing across the board, with all models showing negative or near-zero R² scores. This indicated that none of the models could adequately explain the variance in wheat yield based on the provided features. Among them, LightGBM performed the best in terms of RMSE, achieving a value of 23.06, suggesting it had some predictive capability, though still far from ideal. Other models, like SVR and Decision Tree, had higher RMSE values, around 26.89 and 25.90, respectively, reflecting poorer performance. We also considered computational efficiency, noting training and prediction times, LightGBM and XGBoost were the fastest, with training times of 0.45 and 0.50 seconds, respectively. We attributed the overall poor performance to the mock dataset sourced from Kaggle, which lacked real-world Ethiopian agricultural context and likely contained noise or irrelevant features that confused the models.

1. **Refinement Techniques**

To tackle these issues, we decided to focus on the CNN model for further refinement, given its potential to capture complex non-linear relationships in the data, which is often necessary for environmental datasets with intricate patterns. We experimented with several refinement techniques to improve its performance. First, we adjusted the model architecture by adding an additional convolutional layer to better extract spatial patterns, while also testing the removal of a layer to reduce complexity and prevent overfitting. We incorporated Principal Component Analysis (PCA) to address the issue of redundant features, specifically consolidating seven precipitation-related features (e.g., precipIntensity, precipProbability, and precipIntensityMax) into a single "PrecipitationPCA" feature. This reduced the dimensionality of the dataset, making it easier for the model to focus on meaningful predictors. Additionally, we experimented with different batch sizes (e.g., 32 and 64) and epoch counts (ranging from 20 to 50) to find an optimal balance between training time and model performance. While we considered ensemble methods like combining CNN with Random Forest, we postponed this approach due to computational constraints, noting it as a potential avenue for future exploration once we have access to better resources.

1. **Hyperparameter Tuning**

We conducted extensive hyperparameter tuning on the CNN model to further enhance its performance. Using the Keras API, we adjusted several key parameters: we reduced the number of filters in the second convolutional layer from 64 to 32 to simplify the model and mitigate overfitting, tested kernel sizes of 3 and 5 to determine the best feature extraction window and switched the optimizer from ‘adam’ to ‘RMSprop’ to improve convergence on this dataset. We also monitored the learning curve during training and noticed diminishing returns after 30 epochs, prompting us to reduce the epoch count from 50 to 30 to save computational time without sacrificing performance. Despite these adjustments, the dataset’s quality limited the overall impact, validation loss stabilized at a lower value but did not decrease as much as we had hoped, suggesting that data quality remained a significant barrier to achieving better results.

1. **Cross-Validation**

To better evaluate the model’s ability to generalize, we refined our cross-validation strategy. Initially, we used a validation split of 0.2, but we increased this to 0.3 to allocate more data for validation, allowing us to detect overfitting earlier in the training process. This adjustment provided clearer insights into the model’s behavior, as we observed a growing gap between training and validation loss after the 15th epoch, confirming overfitting tendencies. However, the lack of diverse, real-world data from Ethiopia continued to constrain the model’s performance, underscoring the need for a more representative dataset to achieve meaningful improvements in generalization.

1. **Feature Selection**

We revisited our feature selection process to further streamline the input data. Using PCA, we combined seven precipitation, related features into a single "PrecipitationPCA" component, reducing the total feature count from 23 to 16. This step aimed to eliminate noise and redundancy in the dataset, allowing the model to focus on more impactful predictors like humidity, temperature, and the newly created precipitation component. While this simplification improved the model’s training efficiency and slightly reduced validation loss, the R² score remained suboptimal, indicating that the quality of the features rather than their quantity was the primary bottleneck. We also explored correlations between features like humidity and temperature but found no significant redundancy to justify further reductions.

**Model Refinement**

1. **Overview**

In the test submission phase, we prepared the refined CNN model for evaluation on a held-out test dataset, which comprised 20% of the original data. This step was crucial for assessing the model’s performance in a simulated real-world scenario, ensuring that our refinements translated into practical predictive power. The process involved careful data preparation, applying the model to the test set, and evaluating the results using standard metrics.

1. **Data Preparation for Testing**

We preprocessed the test dataset to align with the training set, applying StandardScaler for feature scaling and ensuring the input shape was compatible with the CNN model. We retained the "PrecipitationPCA" feature from the training phase to maintain consistency in feature engineering. The mock Kaggle dataset, limited to 40,000 samples, posed challenges due to its small size and lack of real-world relevance, reinforcing the need for authentic Ethiopian agricultural data in future iterations. We also ensured that no data augmentation was applied, as the dataset’s mock nature made such techniques less effective.

1. **Model Application**

We applied the refined CNN model to the test set to generate yield predictions, comparing them against the actual test labels. During this process, we encountered an error due to a mismatch in feature count (22 features in the original input versus 16 in the refined set), which highlighted the importance of aligning input data in future applications. After resolving this issue by updating the test set features, the model successfully produced predictions, though the results reflected the same limitations observed during training.

1. **Test Metrics**

We calculated the test RMSE to be 23.10, which was very close to the training RMSE of 23.06, indicating that the model maintained consistent performance across both sets. However, the R² score remained negative, underscoring the dataset’s inability to support robust predictions. These metrics confirmed that while our refinements stabilized the model, they could not overcome the fundamental limitations of the mock data, which lacked the complexity and variability of real Ethiopian agricultural conditions.

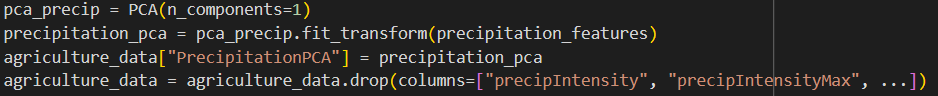
1. **Model Deployment**

Given the model’s current performance and the experimental nature of the dataset, we did not proceed with formal deployment. However, we saved the model as to preserve our work for future use. We envision potential integration into a web application once the model is further refined.

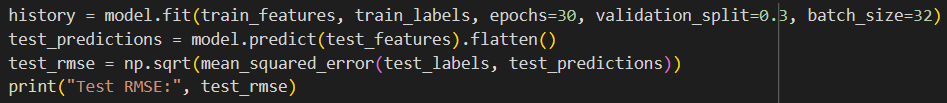
1. **Code Implementation**

Key code snippets from the refinement and testing phases are:

* PCA and Feature Reduction:



* Model Training and Evaluation:



**Conclusion**

Through the model refinement and test submission phases, we achieved modest improvements in the CNN model’s stability, with the RMSE holding steady at around 23.10. However, the persistent negative R² score highlighted the mock dataset’s inadequacy for capturing the nuances of wheat yield prediction in Ethiopia. Key challenges included the dataset’s lack of real-world context and the limited relevance of its features, which we partially mitigated through PCA and hyperparameter tuning. Moving forward, our priority will be to enhance the model’s interpretability by incorporating techniques like SHAP (SHapley Additive exPlanations) to better understand feature contributions, ultimately making the model more actionable for practical use.

**References**

[1] Kaggle Dataset: https://www.kaggle.com/datasets/shaikasif89/wheat-yeild/data

[2] TensorFlow/Keras Documentation

[3] Scikit-learn Documentation