Machine Learning Approaches to “Prediction of Wild Blueberry Yield” Kaggle Competition

Jocelyn McConnon

February 10, 2024

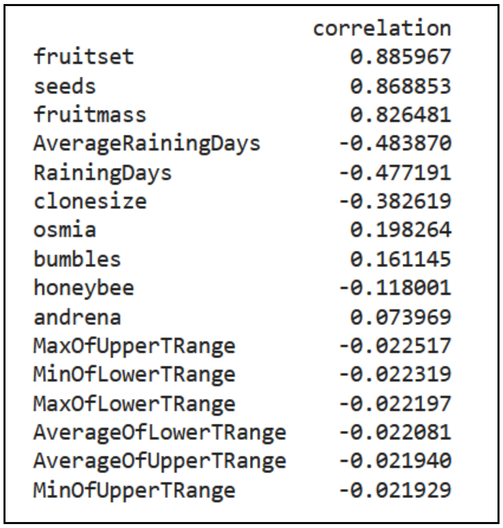
Introduction

Machine learning competitions on platforms like Kaggle offer an invaluable opportunity for data enthusiasts and practitioners to showcase their skills and collaborate with peers. In this report, we delve into various machine learning approaches employed in the context of the Kaggle competition centered around predicting wild blueberry yield.

The competition provides participants with a synthetic dataset comprising features relevant to wild blueberry cultivation.These features encompass a spectrum of environmental, geographical, and agricultural factors that influence the yield of these nutritious berries. At the heart of the competition lies the target variable: wild blueberry yield. This variable represents the quantity of blueberries harvested from a given area of cultivation. Predicting this yield accurately is not only of interest to farmers and agricultural stakeholders but also holds implications for supply chain management, market forecasting, and resource allocation within the agricultural sector.

Participants in the competition are tasked with developing predictive models using the provided dataset and submitting their predictions for evaluation. Submissions are evaluated based on the Mean Absolute Error (MAE), a common metric used to assess the accuracy of regression models. The MAE measures the average absolute difference between the predicted yield and the actual yield across all samples in the test dataset. Achieving low MAE scores indicates a high level of accuracy in predicting wild blueberry yield, thereby demonstrating the effectiveness of the developed machine learning models.

Data Description



```
                     correlation
fruitset                0.885967
seeds                   0.868853
fruitmass               0.826481
AverageRainingDays     -0.483870
RainingDays            -0.477191
clonesize              -0.382619
osmia                   0.198264
bumbles                 0.161145
honeybee               -0.118001
andrena                 0.073969
MaxOfUpperTRange       -0.022517
MinOfLowerTRange       -0.022319
MaxOfLowerTRange       -0.022197
AverageOfLowerTRange   -0.022081
AverageOfUpperTRange   -0.021940
MinOfUpperTRange       -0.021929
```

The dataset for this competition comprises a variety of attributes capturing environmental factors, pollinator densities, and agricultural metrics crucial for understanding and predicting blueberry yields.

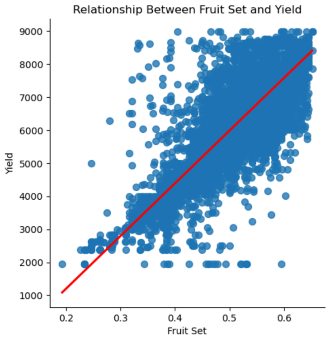
Environmental Factors: The dataset includes attributes related to temperature conditions during the bloom season, such as the highest and lowest recorded daily air temperatures for both upper and lower bands, as well as their corresponding averages. Additionally, the dataset provides information on rainfall patterns, including the total number of rainy days during the bloom season and the average number of raining days throughout the entire season. These environmental factors play a significant role in shaping the growth and development of blueberry plants.

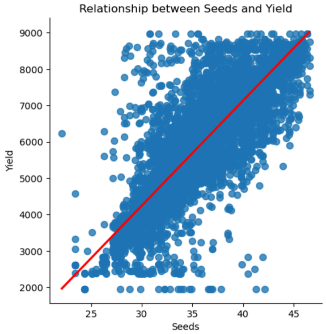
Pollinator Density: Another set of attributes in the dataset pertains to pollinator densities in the blueberry fields. This includes data on the density of honeybees, bumblebees, Andrena bees, and Osmia bees. Pollinators play a crucial role in the fertilization process of blueberry plants, thereby affecting the overall yield.

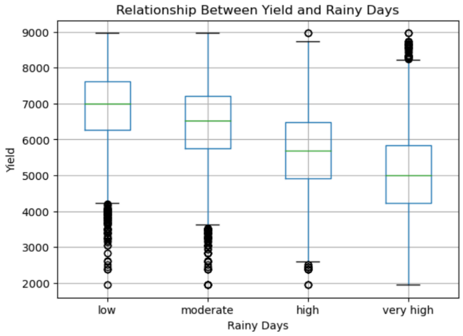
Agricultural Metrics: Lastly, the dataset contains agricultural metrics such as the average blueberry clone size in the field. Understanding the size and health of blueberry clones is essential for assessing the potential yield of a given area of cultivation.

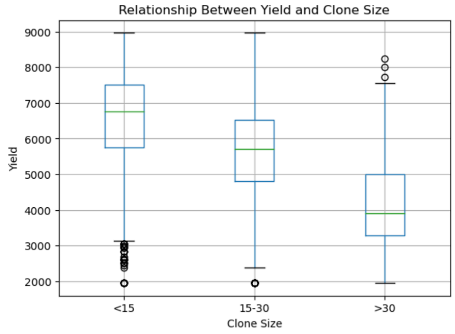
Yield-Related Parameters: In addition to environmental and agricultural factors, the dataset includes attributes directly related to blueberry yield. These include fruitset, fruitmass, seeds, and yield. While fruitset and fruitmass provide insights into the quantity and quality of blueberries produced, seeds may offer information on reproductive success. Yield, the target variable of the competition, represents the quantity of blueberries harvested from a given area of cultivation.

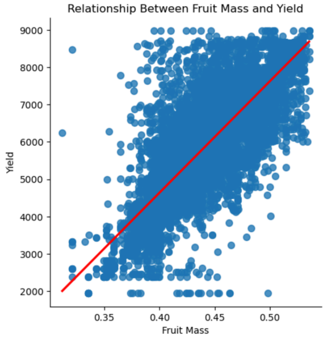
To begin the analysis, an examination was conducted on the five most highly correlated attributes within the dataset. These attributes include fruitset, seeds, fruitmass, AverageRainingDays, and clonesize. In order to gain a deeper understanding of the correlations among these attributes, simple scatter plots and boxplots were plotted to illustrate their relationships. These visualizations offer insights into the relationship between the attributes and their influence on the target variable, wild blueberry yield.











Methodology

To compare the performance of different models and techniques, I applied consistent feature engineering and selection methods. This ensures ease of comparison across results. For feature engineering, I introduced a new attribute called "fruit_seed," calculated as the product of fruit_set and seeds. This addition improved model performance and exhibited correlation with the results. Regarding feature selection, I examined the absolute values of the correlation matrix and retained attributes with correlations exceeding a threshold of 0.5. The final set of attributes used includes: MinOfUpperTRange, AverageOfUpperTRange, MaxOfLowerTRange, MinOfLowerTRange, AverageOfLowerTRange, AverageRainingDays, fruitmass, seeds, fruit_seed.

Model Development

The following models were compared:

XGBRegressor: The data was split using train_test_split with a test_size of 0.2, allocating 20% of the data to the test set. Random state was set to 42. All parameters for this model were left as default.

VotingRegressor Ensemble (LGBMRegressor + CatBoostRegressor) (Vanvinckenroye, 2023): To train this model, the data was split using train_test_split with a test_size of 0.2. Random state was set to 42. Custom parameters were set for the LGBMRegressor model: n_estimators = 250, num_leaves = 57, learning_rate = 0.04, and boosting_type = Gradient Boosting Decision Tree. For the CatBoostRegressor model, n_estimators was set to 250, learning_rate was set to 0.09, and grow_policy was set to lossguide with loss_function set to MAE. The two models were combined using VotingRegressor.

Equal Weights Ensemble (GradientBoostingRegressor + RandomForestRegressor): The equal weights for this ensemble were set to 0.5 since two models were being used. Parameters for the GradientBoostingRegressor model were set as follows: n_estimators = 200, max_depth = 8, and learning_rate = 0.4. For the RandomForestRegressor, n_estimators were set to 150 and max_depth was set to 10. Each model was trained on the split data, and the final predictions used in the submission and evaluation were taken with equal weights, i.e., prediction_gb * 0.5 + prediction_rf * 0.5.

Performance-Based Weights Ensemble (GradientBoostingRegressor + RandomForestRegressor) (Aguilar, 2023): The GradientBoostingRegressor model parameters were set as follows: n_estimators = 200, max_depth = 8, and learning_rate = 0.04. Parameters for the RandomForestRegressor were set to n_estimators = 150 and max_depth = 10. This model training process was the most complex, involving 10-fold validation to evaluate the models' performance. On each fold, the data was randomly split, and the models were fit to the subset data. An ensemble model was created using weights derived from the two models, and the ensemble prediction was evaluated using MAE. The final weights used for the models were a 10-d array of the ensemble MAE scores, which were used to calculate the relative performance of each fold, and the predictions for each fold were weighted accordingly.

For the full list of parameters used in each model, please refer to the source code for this report.

Results

The best performing model was the Performance-Based Weights Ensemble, consisting of GradientBoostingRegressor and RandomForestRegressor. This method, utilizing results from 10 different folds, represented the most advanced approach implemented in the analysis. A simpler alternative was the VotingRegressor Ensemble, combining LGBMRegressor and CatBoostRegressor. Despite its simplicity, this model demonstrated competitive performance comparable to the Performance-Based Weights Ensemble. In contrast, the XGBRegressor exhibited the least efficiency in prediction among the models tested.

Model	MAE	10-fold cross-validation mean
XGBRegressor	373.703	yes
VotingRegressor Ensemble, LGBMRegressor + CatBoostRegressor	345.780	yes
Equal weights Ensemble, GradientBoostingRegressor + RandomForestRegressor	353.081	no
Performance based weights Ensemble, GradientBoostingRegressor + RandomForestRegressor	344.251	yes

The competition public leaderboard score achieved was 338.01074, securing a position of 196 out of 1875 participants, placing in the top 11%. It's noteworthy that the public leaderboard is evaluated on a new testing dataset.

Discussion

I conducted experiments with grouping yield values and averaging other numerical attributes but found no improvement in model performance; consequently, this approach was not implemented. Moreover, utilizing the full set of attributes did not enhance model performance; therefore, I chose a threshold of 0.5 to select features. Moving forward, I believe that more robust feature engineering is necessary. Standardizing certain numerical attributes could potentially enhance model performance. Additionally, future improvements to this methodology might involve binning attributes and converting numerical values into categorical attributes.

For the XGBRegressor model, performance can potentially be improved through hyperparameter tuning. As the parameters were left as default values, there may be opportunities for enhancements in performance. Regarding the VotingRegressor Ensemble, LGBMRegressor + CatBoostRegressor method, while seemingly precise, the parameters used in model development were not determined through hyperparameter tuning. Given more time, I believe this method could benefit from more optimally chosen parameters.

Overall, a decision should have been made in the methodology regarding hyperparameter tuning or leaving all parameters as default. The randomness of the value selection makes it challenging to accurately compare performance across different methods.

Furthermore, some of the more advanced methods may suffer from overfitting, as the models may become too complex for the small, simple dataset, especially considering the limited number of attributes used. Another possible extension could involve running an even simpler model on the data to compare performance effectively.

Conclusion

The dataset consisted of basic agricultural and environmental factors known to influence wild blueberry yield. It is evident that several attributes, such as rainy days, seeds, and fruitmass, exhibit correlations with the target variable, yield. This competition provided a valuable opportunity to explore various models and their performance on the synthetic dataset. Further emphasis on feature engineering and selection, along with meticulous hyperparameter tuning, could potentially enhance the individual performances of these models.

References

Kaggle. 2023. “Playground Series S3E14 Competition”. Retrieved from https://www.kaggle.com/competitions/playground-series-s3e14.

Vanvinckenroye, Jonas. 2023. “S3-E14 | EDA | Model comparison”. Retrieved from https://www.kaggle.com/code/jonasvanvinckenroye/s3-e14-eda-model-comparison.

Aguilar, Oscar. 2023. “PS-S3-Ep14 | EDA 📊 | Modeling + Submission 🚀”. Retrieved from https://www.kaggle.com/code/oscarm524/ps-s3-ep14-eda-modeling-submission.