Regression Analysis of Predicting Deviation from highest grade Sweet Potato

Rashmi Datta
NC State University
Raleigh, NC, USA
rdatta2@ncsu.edu

***Abstract*—Sweet potatoes as a crop is now viewed as a high business asset due to its versatility and high nutritional value. However, variety classification on this commodity is considerably significant to ensure the quality of products before reaching production. However, its production has been declining, despite the increasing demand from the fresh roots market due to various factors such as mutation, viruses, and other pathogens in sweet potato diseases. Hence, predicting the deviation from the highest grade sweet potato such US#1 is important to ensure the quality of products before reaching production. This helps farmers, specialists, and decision-makers to efficiently qualify sweet potatoes of different types. This study presents a comparative analysis between different sets of data used to grade high quality sweet potato using various regression models. The results of the study show that the Random Forest regression model with GridSearchCV and Cross-Validation performs the best with an average mean-square error of 265.51. Practical implications and research directions are presented.**

***Index Terms*—Regression, Random Forest, Mean-square error, Deviation from US#1, user-defined features, extracted features**

I. Introduction

In many countries, sweet potato is a significant staple and urgent food. In worldwide food production, sweet potato is seventh and the third- largest sought-after potato variety. It is a vital ingredient in animal diets on many small farms. To date, sweet potato is continuously growing, finding a part in elite urban diets, and used for processed products. The goal of this project is developing new data analysis tools for streamlining the North Carolina Sweet Potato industry. A nexus point of all crop value is located at sorting facilities, where produce is sorted into value-added categories. These physical characteristics will be linked to up-stream provenance data (field location, weather data, management practices) and down-stream value (consumer preferences, storage life, etc.) to develop new management practices that maximize value.

Through this study we have tried to find out the best means to objectively and subjective grade sweet potatoes w.r.t their deviation from US#1. We have successfully collected sweet potato data from three sources:

- User-defined features are obtained from a GUI (which will first select the role - breeder, grower, researcher, student and then grade the SP on the following parameters): Roundness, Tailedness, Curviness, Blockiness, and Deviation from US#1 (each on a scale from 0% to 100%).
- 3D reconstruction extracted features which include : AxialLength, TipLength, Curvature, MaxDiameter, LWRatio, TailLength, TailPct, Shape, BodyLength, TailBodyRatio, Volume, AverageCrossSectionRadius, weight.
- Templating data extracted features which includes: firstorder count, tail count, residerror count, roothairs count, sphere errors count, largeresiderror count, area(pixel), length(pixel), width(pixel), length(inches), width(inches), Weight(oz), Weight(g).

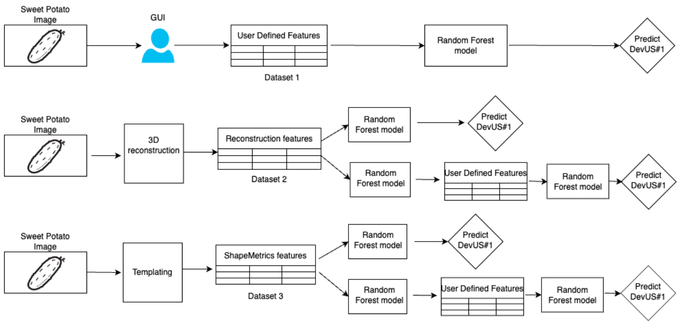


Fig. 1. Flow of our analysis to predict Deviation from US#1

Our problem statement is broken down into regression analysis in three parts:

- Predict Deviation from US#1 from user-defined features and assess its performance.
- Describe each of the user-defined features in terms of combination of the extracted features (both 3D reconstruction and templating datasets)
- Predict Deviation from US#1 from extracted features and assess its performance.

Our target prediction helps us in deciding how well the user -defined features do w.r.t to the extracted features to predict the deviation from US#1 of sweet potatoes.

II. Literature Review

We have used five regression models to test our hypothesis: Lasso, Ridge, Elastic Net, Random Forest and Support Vector Regression(SVR)

A. Lasso Regression

Lasso is a linear regression technique that focuses on modeling the linear relationship between input features and the target variable with regularization. It is commonly used for regression tasks, where the goal is to find the best-fitting

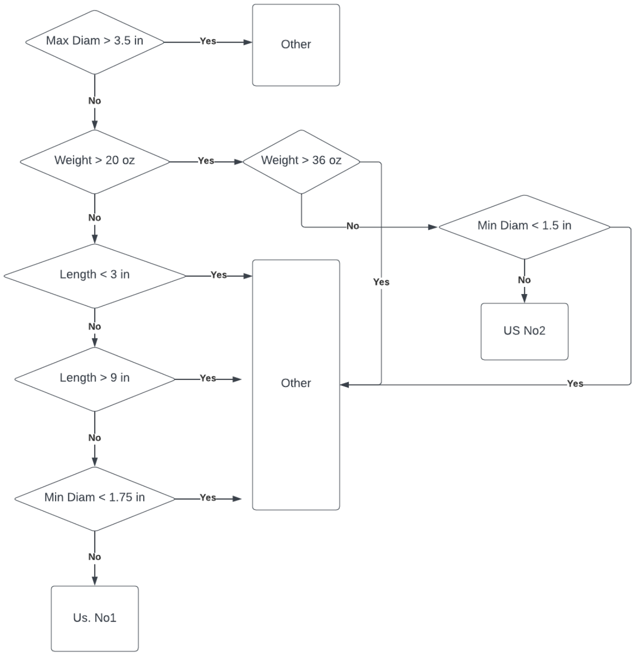


Fig. 2. Images are classified as US#1 by if-else logic using extracted features, only then US#1 images are submitted for US#1-deviation (see chart above)

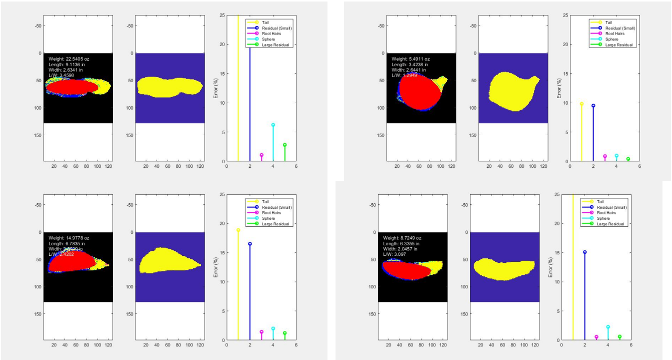


Fig. 3. Samples from the templating dataset indicating various shapemetrics while evaluating sweet potato images.

linear equation to predict a continuous target variable. The objective of lasso regression is to minimize the sum of two components: Residual Sum of Squares (RSS) and Penalty Term (also called the "L1 regularization term"). The second term in the below equation controls the strength of the penalty.

Objective = RSS + $\lambda \sum_{i=1}^{n} |\beta_i|$
where:
n is the number of features
β_i are the coefficients associated with each feature
λ is the regularization parameter that controls the trade-off between fitting the data well and keeping the coefficients small.

The penalty term helps prevent overfitting by shrinking the coefficients of features, reducing their impact on the model. Thus, Lasso helps identify the most relevant features, reducing the complexity of the model and improving its interpretability.

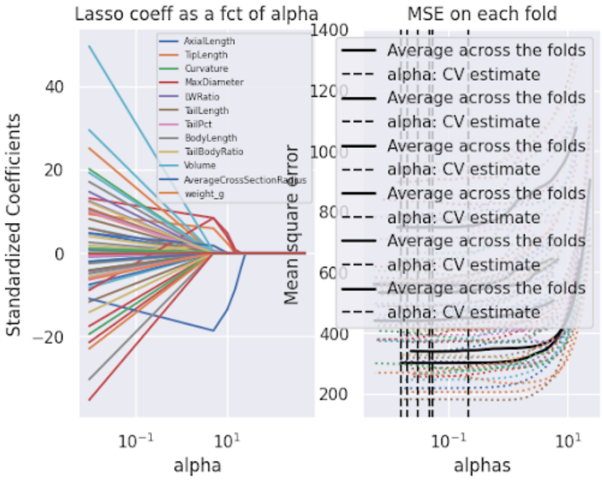


Fig. 4. Lasso regression results obtained with our initial model

B. *Ridge Regression*

Ridge Regression is a linear regression technique that adds L2 regularization to mitigate multi collinearity (high correlation between independent variables/features) and prevents over fitting. Ridge Regression aims to minimize a modified version of the ordinary least squares (OLS) loss function.

Objective = $\sum_{i=1}^{n}(y_i - y_i^{'})^2 + \lambda \sum_{i=j}^{p} \beta_j^2$
where:
n is the number of data points
p is the number of features (independent variables)
y_i is the observed target value for the i-th data point
$y_i^{'}$is the predicted target value for the i-th data point
β_j is the coefficient (weight) associated with the j-th feature
λ is the regularization hyperparameter that controls the strength of penalty term.

The second term in the objective function is the regularization term(L2 penalty term). The purpose of this term is to add a penalty for large coefficient values. The larger the coefficients, the higher the penalty, which discourages the model from assigning too much importance to any single feature. The choice of the regularization parameter λ is crucial. A small λ allows the model to fit the training data more closely, while a large λ increases the penalty on coefficient magnitudes, encouraging simpler models. Cross-validation is often used to find the optimal λ that minimizes prediction error on a validation set. Ridge Regression can be solved using various optimization techniques. One common approach is to use gradient descent to minimize the objective function.

C. *Elastic Net Regression*

Elastic Net Regression is a regression technique that combines the principles of both Ridge Regression and Lasso Regression. It's used to address the limitations of these two techniques while benefiting from their strengths. Elastic Net introduces both L1 (Lasso) and L2 (Ridge) regularization terms into the linear regression objective function, as seen below.

Objective = $\sum_{i=1}^{n}(y_i - y_i^{'})^2 + \lambda_1 \sum_{i=j}^{p} \beta_j + \lambda_2 \sum_{i=j}^{p} \beta_j^2$

Where λ_1 and λ_2 are regularization parameters, hyper parameters that control the strengths of the L1 (Lasso) and L2 (Ridge) penalty terms, respectively. The first term $\lambda_1 \sum_{i=j}^{p} \beta_j$ is the L1 regularization term, which similar to the Lasso penalty, encourages sparsity in the model, effectively performing feature selection by setting some coefficients to exactly zero.

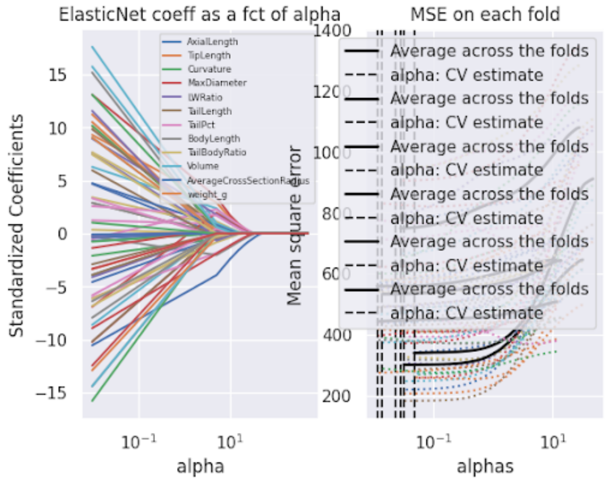


Fig. 5. Elastic Net regression results obtained with our initial model

The second term $\lambda_2 \sum_{i=j}^{p} \beta_j^2$ is the L2 regularization term, which similar to the Ridge penalty, discourages large coefficient values and helps mitigate multi collinearity.

Cross-validation is often used to find the combination of λ_1 and λ_2 that minimizes prediction error on a validation set. The choice of parameters depends on the problem and the trade-off between sparsity (L1) and coefficient shrinkage (L2) you desire.

In summary, this technique is particularly useful when we have a dataset with correlated features, want to perform feature selection, and wish to control over fitting in a linear regression model.

D. *Random Forest Regression*

Random Forest Regression combines multiple decision trees to make predictions, resulting in a robust and accurate model. It is an ensemble learning technique used for regression tasks, which involve predicting continuous target variables.

Each tree in the ensemble is built independently and is referred to as a ”base tree” or ”base learner.” Random Forest uses a technique called Bootstrap Aggregating or Bagging. For each tree in the ensemble, a random sample (with replacement) is drawn from the original dataset. This creates a diverse set of training datasets for each tree. Random Forest also employs feature selection randomness. When building each decision tree, only a random subset of the features (predictor variables) is considered at each split point. This helps in reducing the correlation among trees.

To make a prediction, each tree in the ensemble independently produces an output (a regression value). The final prediction from the Random Forest is often the average (or weighted average) of the predictions from all the trees. This regression tool is highly robust to overfitting because:

- It combines multiple trees, which collectively generalize better.
- It can handle large datasets with many features.
- It automatically handles missing values and outliers.
- It provides feature importance scores, indicating the contribution of each feature to the model’s predictions.

E. *Support Vector Regression*

Support Vector Regression (SVR) is a regression technique that extends the principles of Support Vector Machines (SVMs) to the task of regression, which involves predicting a continuous target variable.

SVR is particularly useful when dealing with complex data distributions, non-linear relationships, and data with outliers. The primary goal of SVR is to find a function f(x) that can make accurate predictions for a continuous target variable y based on input features x. SVR aims to minimize the error between the predicted values f(x) and the actual target values y, while also considering a tolerance margin or ”epsilon tube”(ϵ-tube) around the regression line. Data points that fall within this tube are not penalized, and their errors are considered acceptable. Data points outside the epsilon tube contribute to the loss function and are penalized based on their distance from the tube. Support vectors in SVR are data points that fall on the border of the epsilon tube or within the tube itself. These are most influential in determining the position and orientation of the regression line.

SVR introduces a regularization parameter C, which controls the trade-off between maximizing the margin (large C) and minimizing the error (small C). Smaller C values allow for a larger margin but may tolerate more errors, while larger C values prioritize reducing errors, potentially leading to a smaller margin. SVR uses a loss function that encourages the predicted values to be as close to the actual values as possible while staying within the epsilon tube. This function includes a term for minimizing the error and a term for maximizing the margin.

III. Methods

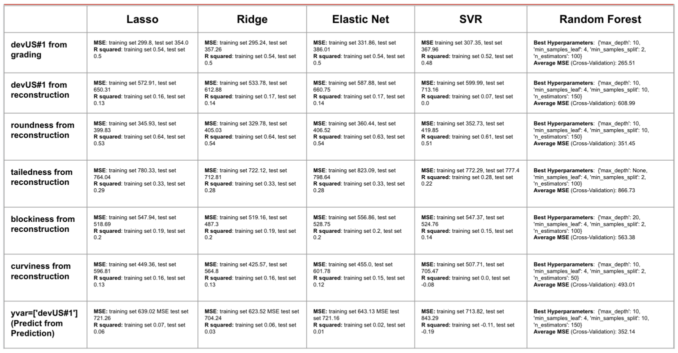
Our research works with the data resulting from the high-throughput imagery approach to sweet potato shape and size detection (resulting from Haque et al. 2021,Ref [1]), combined with our subjective user scoring project, put together by Daniel Perondi. The shapemetrics dataset is compiled by Michael Kudenov’s results of the templating algorithm.

The objective, image-based data is in the dataset called ”reconstruction”. Features obtained include: AxialLength, TipLength, Curvature, MaxDiameter, LWRatio (length/width ratio), TailLength, TailPct (tail%), Shape, BodyLength, TailBodyRatio, Volume, AverageCrossSectionRadius, weight-g (calculated by volume with coefficient for density), weight-oz (converted from weight-g).

The subjective, classification data is in the dataset called ”grading”. Features obtained include: classified-image-id, roundness tailedness, blockiness, curviness, deviation-from-usno1, graded (yes/no), image-id, image-name (hashed), usdaclass-name,axial-length, max-diameter, lw-ratio, weight-oz, user-id, user-role (breeder, student, researcher)

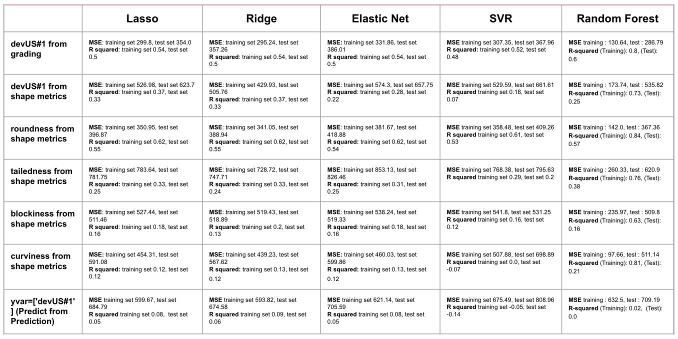
The second objective, templating data is in the dataset called ”shapemetrics”. Features obtained include: firstorder count, tail count, residerror count, roothairs count, sphere errors count, largeresiderror count, area(pixel), length(pixel), width(pixel), length(inches), width(inches), Weight(oz), Weight(g).

In the following tables, we have used various metrics (such as MSE and R-squared) to evaluate the performances of each of the five regression models and then compared their results to find our best performing/ champion model for further analysis.



	Lasso	Ridge	Elastic Net	SVR	Random Forest
devUS#1 from grading	**MSE**: training set 299.8, test set 354.0 **R squared**: training set 0.54, test set 0.5	**MSE**: training set 295.24, test set 357.26 **R squared**: training set 0.54, test set 0.5	**MSE:** training set 331.86, test set 386.01 **R squared**: training set 0.54, test set 0.5	**MSE** training set 307.35, test set 367.96 **R squared:** training set 0.52, test set 0.48	**Best Hyperparameters**: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100} **Average MSE** (Cross-Validation): 265.51
devUS#1 from reconstruction	**MSE**: training set 572.91, test set 650.31 **R squared**: training set 0.16, test set 0.13	**MSE**: training set 533.78, test set 612.88 **R squared**: training set 0.17, test set 0.14	**MSE**: training set 587.88, test set 660.75 **R squared**: training set 0.17, test set 0.14	**MSE:** training set 599.99, test set 713.16 **R squared:** training set 0.07, test set 0.0	**Best Hyperparameters**: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 150} **Average MSE** (Cross-Validation): 608.99
roundness from reconstruction	**MSE**: training set 345.93, test set 399.83 **R squared**: training set 0.64, test set 0.53	**MSE**: training set 329.78, test set 405.03 **R squared**: training set 0.64, test set 0.54	**MSE**: training set 360.44, test set 406.52 **R squared**: training set 0.63, test set 0.54	**MSE**: training set 352.73, test set 419.85 **R squared:** training set 0.61, test set 0.51	**Best Hyperparameters**: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 150} **Average MSE** (Cross-Validation): 351.45
tailedness from reconstruction	**MSE**: training set 780.33, test set 764.04 **R squared**: training set 0.33, test set 0.29	**MSE**: training set 722.12, test set 712.81 **R squared**: training set 0.33, test set 0.28	**MSE**: training set 823.09, test set 798.64 **R squared**: training set 0.33, test set 0.28	**MSE:** training set 772.29, test set 777.4 **R squared:** training set 0.28, test set 0.22	**Best Hyperparameters**: {'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100} **Average MSE** (Cross-Validation): 866.73
blockiness from reconstruction	**MSE**: training set 547.94, test set 518.69 **R squared**: training set 0.19, test set 0.2	**MSE**: training set 519.16, test set 487.3 **R squared**: training set 0.19, test set 0.2	**MSE**: training set 556.86, test set 528.75 **R squared**: training set 0.2, test set 0.2	**MSE:** training set 547.37, test set 524.76 **R squared:** training set 0.15, test set 0.14	**Best Hyperparameters**: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100} **Average MSE** (Cross-Validation): 563.38
curviness from reconstruction	**MSE**: training set 449.36, test set 596.81 **R squared**: training set 0.16, test set 0.13	**MSE**: training set 425.57, test set 564.8 **R squared**: training set 0.16, test set 0.13	**MSE**: training set 455.0, test set 601.78 **R squared**: training set 0.15, test set 0.12	**MSE:** training set 507.71, test set 705.47 **R squared:** training set 0.0, test set -0.08	**Best Hyperparameters**: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50} **Average MSE** (Cross-Validation): 493.01
yvar=['devUS#1'] (Predict from Prediction)	**MSE:** training set 639.02 MSE test set 721.26 **R squared:** training set 0.07, test set 0.06	**MSE:** training set 623.52 MSE test set 704.24 **R squared:** training set 0.06, test set 0.03	**MSE:** training set 643.13 MSE test set 721.16 **R squared:** training set 0.02, test set 0.01	**MSE:** training set 713.82, test set 843.29 **R squared:** training set -0.11, test set -0.19	**Best Hyperparameters**: {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 150} **Average MSE** (Cross-Validation): 352.14

Fig. 6. Performance Results tabulated across all models using reconstruction and user-defined features



	Lasso	Ridge	Elastic Net	SVR	Random Forest
devUS#1 from grading	**MSE**: training set 299.8, test set 354.0 **R squared**: training set 0.54, test set 0.5	**MSE**: training set 295.24, test set 357.26 **R squared**: training set 0.54, test set 0.5	**MSE:** training set 331.86, test set 386.01 **R squared**: training set 0.54, test set 0.5	**MSE** training set 307.35, test set 367.96 **R squared:** training set 0.52, test set 0.48	**MSE** training : 130.64, test : 286.79 **R-squared** (Training): 0.8, (Test): 0.6
devUS#1 from shape metrics	**MSE**: training set 526.98, test set 623.7 **R squared**: training set 0.37, test set 0.33	**MSE**: training set 429.93, test set 505.76 **R squared**: training set 0.37, test set 0.33	**MSE**: training set 574.3, test set 657.75 **R squared**: training set 0.28, test set 0.22	**MSE** training set 529.59, test set 661.61 **R squared** training set 0.18, test set 0.07	**MSE** training : 173.74, test : 535.82 **R-squared** (Training): 0.73, (Test): 0.25
roundness from shape metrics	**MSE**: training set 350.95, test set 396.87 **R squared:** training set 0.62, test set 0.55	**MSE**: training set 341.05, test set 388.94 **R squared:** training set 0.62, test set 0.55	**MSE**: training set 381.67, test set 418.88 **R squared:** training set 0.62, test set 0.54	**MSE** training set 358.48, test set 409.26 **R squared** training set 0.61, test set 0.53	**MSE** training : 142.0, test : 367.36 **R-squared** (Training): 0.84, (Test): 0.57
tailedness from shape metrics	**MSE**: training set 783.64, test set 781.75 **R squared:** training set 0.33, test set 0.25	**MSE**: training set 728.72, test set 747.71 **R squared:** training set 0.33, test set 0.24	**MSE**: training set 853.13, test set 826.46 **R squared:** training set 0.31, test set 0.25	**MSE** training set 768.38, test set 795.63 **R squared** training set 0.29, test set 0.2	**MSE** training : 260.33, test : 620.9 **R-squared** (Training): 0.76, (Test): 0.38
blockiness from shape metrics	**MSE**: training set 527.44, test set 511.46 **R squared**: training set 0.18, test set 0.16	**MSE**: training set 519.43, test set 518.89 **R squared**: training set 0.2, test set 0.13	**MSE**: training set 538.24, test set 519.33 **R squared**: training set 0.18, test set 0.16	**MSE** training set 541.8, test set 531.25 **R squared** training set 0.16, test set 0.12	**MSE** training : 235.97, test : 509.8 **R-squared** (Training): 0.63, (Test): 0.16
curviness from shape metrics	**MSE:** training set 454.31, test set 591.08 **R squared:** training set 0.12, test set 0.12	**MSE:** training set 439.23, test set 567.62 **R squared:** training set 0.13, test set 0.12	**MSE:** training set 460.03, test set 599.86 **R squared:** training set 0.13, test set 0.12	**MSE** training set 507.88, test set 698.89 **R squared** training set 0.0, test set -0.07	**MSE** training : 97.66, test : 511.14 **R-squared** (Training): 0.81, (Test): 0.21
yvar=['devUS#1'] (Predict from Prediction)	**MSE** training set 599.67, test set 684.79 **R squared** training set 0.08, test set 0.05	**MSE** training set 593.82, test set 674.58 **R squared** training set 0.09, test set 0.06	**MSE** training set 621.14, test set 705.59 **R squared** training set 0.08, test set 0.05	**MSE** training set 675.49, test set 808.96 **R squared** training set -0.05, test set -0.14	**MSE** training : 632.5, test : 709.19 **R-squared** (Training): 0.02, (Test): 0.0

Fig. 7. Performance Results tabulated across all models using shape metrics and user-defined features

IV. Results

The Random Forest model has outperformed the other models in terms of performance (MSE and R-square). The following figures (Fig.7, Fig.8, Fig.9, Fig.10, Fig.11) depict the results from our best performing model to address our problem statement.

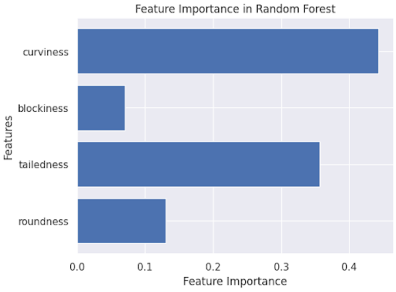


Fig. 8. The feature importance obtained for devUS#1 from each user-graded metric

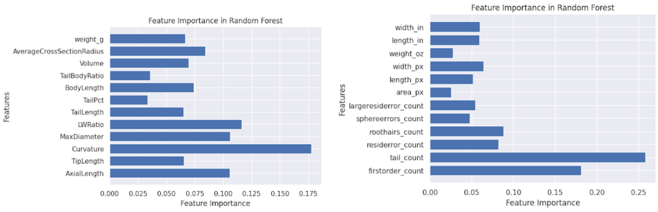


Fig. 9. The feature importances obtained for devUS#1 from reconstruction data and templating data respectively

V. Discussion

Our best performing model Random Forest with GridSearchCV and Cross Validation gave the following results:

- Mean Squared Error for Deviation from US#1 from User-defined features is observed to be 265.51.
- The Mean Squared Error for Deviation from US#1 from reconstruction features is obtained as 608.99.
- The Mean Squared Error from prediction for reconstruction features from user-defined features is 352.14. Thus, the percentage improvement in Mean Squared Error after incorporating user-defined features in reconstruction features = 42.17%.
- The Mean Squared Error for Deviation from US#1 from shape metrics is obtained as 514.18.
- The Mean Squared Error from prediction for shape metrics from user-defined features is 305.73. Therefore, percentage improvement in Mean Squared Error after incorporating user-defined features in shape metric features = 40.54%).

Figure 12 concisely summarizes the above results and their interpretations.

VI. Conclusion:

After assessing the performance across the three sets of data, the user-defined (grading) features performed the best in predicting the deviation from US No.1 of a sweet potato and were the most accurate amongst the three. As we can see from Fig.12, the shape metrics/templating data gave better results as compared to the the reconstruction data and observed a much higher improvement in predictions after Random Forest regression when each of the user-graded features were defined in terms of the objective datasets.

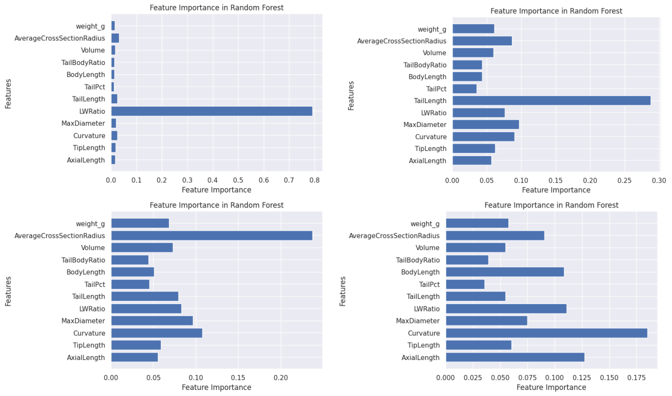


Fig. 10. The feature importances obtained for each user-graded metric (roundness, tailedness, blockiness, curviness) from reconstruction data respectively

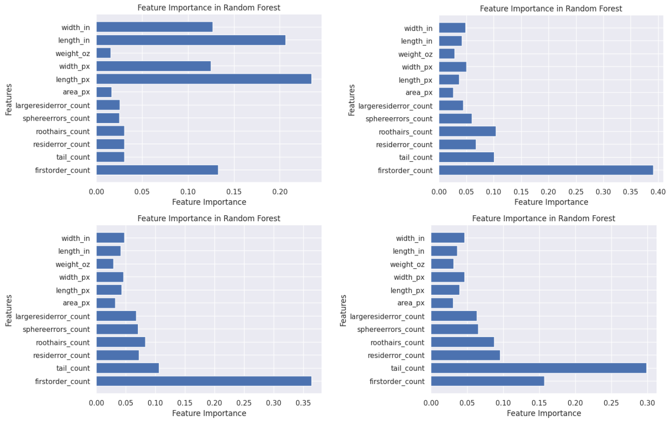


Fig. 11. The feature importances obtained for each user-graded metric (roundness, tailedness, blockiness, curviness) from templating data respectively

VII. Future Works:

The dataset could be biased where a particular user may not have entered the information in the survey from the GUI sincerely or different users might just have very different perceptions of the idea of US#1 sweet potato. For example, 'tail' being confused with other error metrics. In such a case, we can modify the dataset instead of or along with a better regression model, such a neural network, to obtain the most optimal results. Moreover, similar to relative grading, we can implement relative scaling on the user-defined data to deal with variation in human scope. In such a case, we would consider the best estimated sweet potato according to the user and scale the rest of the data w.r.t. to that. Use of auto-encoders to generate more relevant features of a sweet potato from its image is another possibility we plan to explore in order to obtain better results.

References

[1] Samiul Haque, Edgar Lobaton, Natalie Nelson, G. Craig Yencho, Kenneth V. Pecota, Russell Mierop, Michael W. Kudenov, Mike Boyette, Cranos M. Williams. Computer vision approach to characterize size and shape phenotypes of horticultural crops using high-throughput imagery https://doi.org/10.1016/j.compag.2021.106011

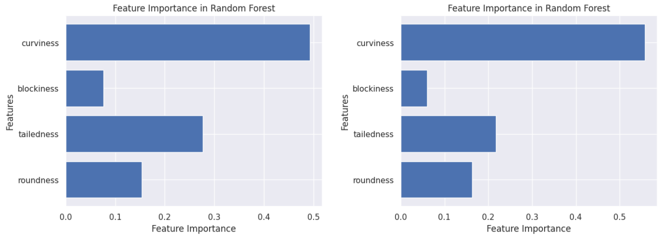


Fig. 12. The predict from prediction results of feature importances obtained for devUS#1 from reconstruction data and templating data respectively

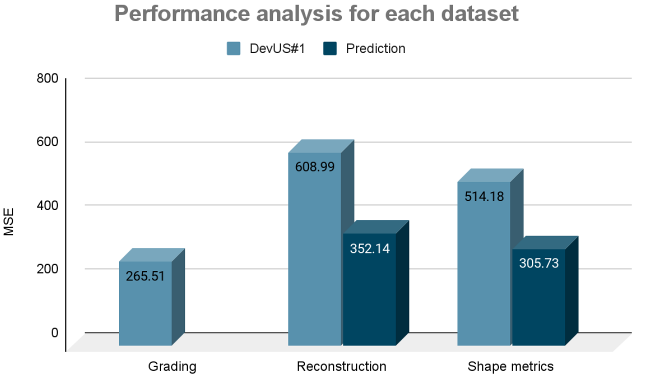


Fig. 13. Comparing performances with Random Forest model across each dataset

[2] USDA Agricultural Marketing Service Sweetpotatoes Grades and Standards https://www.ams.usda.gov/grades-standards/sweetpotatoes-grades-and-standards

[3] SweetAPPS Grading via Templating https://lucid.app/lucidspark/3e54b17c-5122-4b0a-97f2-653804df3fb5/edit?invitationId=inv_364e1d54-7050-4d6d-9fd7-c7622bea4d88&page=0_0

[4] Dexter Mercurio, Alexander A Hernandez. Classification of Sweet Potato Variety using Convolutional Neural Network. https://www.researchgate.net/publication/337502073_Classification_of_Sweet_Potato_Variety_using_Convolutional_Neural_Network

VIII. Source code

The source code for this project is available at : https://github.com/RashmiD25/sweetapps_regression