MSCI 546: Predicting Blueberry Yield

Team 3: Carina Chiu, Rawaha Nakhuda, Gillian Tsoi, Troy Zada

Data: Wild Blueberry Yield

Data Description

- 16 features with over 15,000 rows
- Continuous numerical data
- Labelled

Factors Affecting Yield

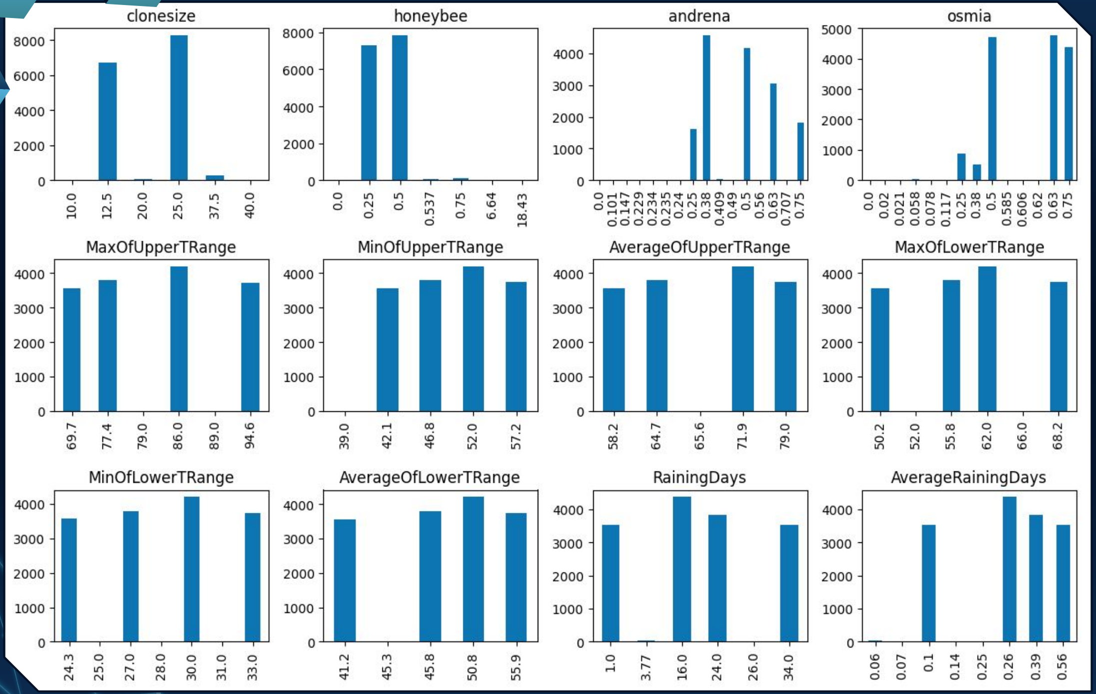
- Bush size, seeds, etc.
  - clonesize
  - seeds
- Pollination
  - honeybee
  - bumbles
- Weather
  - RainingDays
  - AverageOfLowerTRange

https://www.kaggle.com/datasets/shashwatwork/wild-blueberry-yield-prediction-dataset

EDA: Feature Insights

	clonesize	honeybee	bumbles	andrena	osmia	MaxOfUpperTRange	MinOfUpperTRange
count	15289.000000	15289.000000	15289.000000	15289.000000	15289.000000	15289.000000	15289.000000
mean	19.704690	0.389314	0.286768	0.492675	0.592355	82.169887	49.673281
std	6.595211	0.361643	0.059917	0.148115	0.139489	9.146703	5.546405
min	10.000000	0.000000	0.000000	0.000000	0.000000	69.700000	39.000000
50%	25.000000	0.500000	0.250000	0.500000	0.630000	86.000000	52.000000
max	40.000000	18.430000	0.585000	0.750000	0.750000	94.600000	57.200000

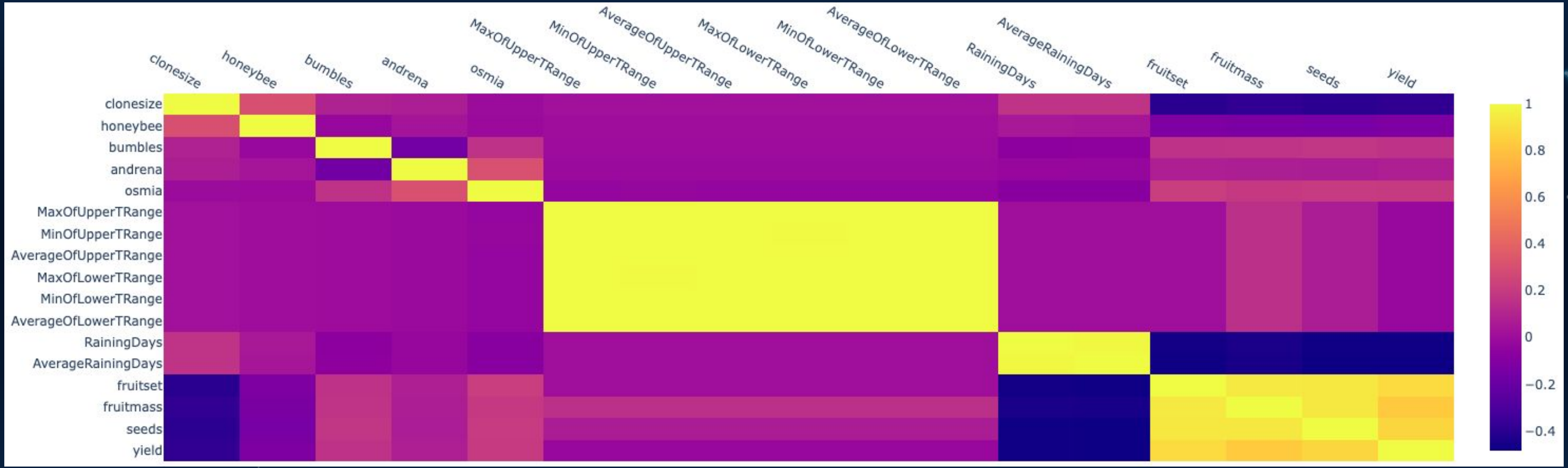
- Features take unique range of values
- No features are missing entries
- Emphasis on normalization to standardize the differing feature distributions



EDA: Feature Distribution

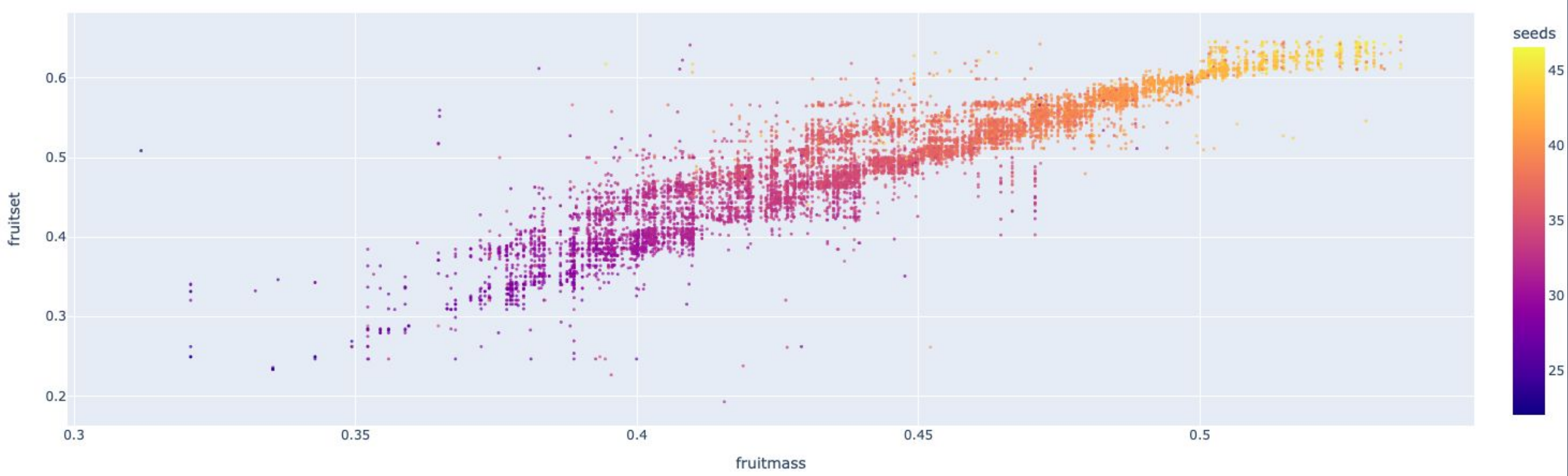
- Highly skewed features, with most values concentrated around a subset of outcomes
- Value imbalance may have limited predictive power on the yield

EDA: Feature Correlations



- High correlation between fruit size features, fruitset, fruitmass, and seeds
- Relatively low correlation between all other combination of features
- Outlines potential feature impacts on predicting the yield

EDA: Highly Correlated Features



- Linear relationship between fruitset, fruitmass, and seeds
- Concentrated clusters when seeds is greater than 40

Task and Metrics

Regression, supervised learning

- **Mean Absolute Error** — For ranking model performance, treating all errors equally
- **Mean Squared Error** — Average of the squares, amplifying larger errors
- **Mean Absolute Percentage Error** — Average percentage difference of values
- **R^2 Score** — Proportion of variance in the target

Regression Models

01

Baseline

Simple linear regression model using all features

02

Ridge Regression

Modified linear regression model applying an L2 penalty

03

Random Forest

Combination of multiple decision trees outputting mean prediction

04

Gradient Boosting

Ensemble tree-based regression model

05

Multilayer Perceptron

Feedforward neural network model

Baseline

Model

- Linear Regression

Methodology

- Simple linear regression with all features

Ridge Regression

	param_alpha	mean_test_score	std_test_score	rank_test_score
0	0.00001	-374.745155	7.734414	5
1	0.0001	-374.745002	7.734283	4
2	0.001	-374.743479	7.732973	3
3	0.01	-374.729939	7.720049	2
4	0.1	-374.722170	7.608128	1
5	1.0	-378.124874	6.918821	6
6	10.0	-400.011216	5.681669	7
7	100.0	-424.416364	5.909648	8
8	1000.0	-431.279908	6.282823	9
9	10000.0	-444.420732	5.792642	10
10	100000.0	-583.172633	5.805182	11

Model

- Minimizes regularized residual sum of squares
- Adds L2 penalty to minimize value of weights

Hyperparameters

- Tuned alpha using grid search

Final Hyperparameters	
Alpha	0.1

Random Forest

Model

- Combination of multiple decision trees, outputting mean prediction of individual trees

Normalization

- Normalized feature data using StandardScalar

Hyperparameters

- Tuned hyperparameters using grid search

Decision at Nodes

- True → Follow left branch
- False → Follow right branch
- Repeats until leaf node is reached

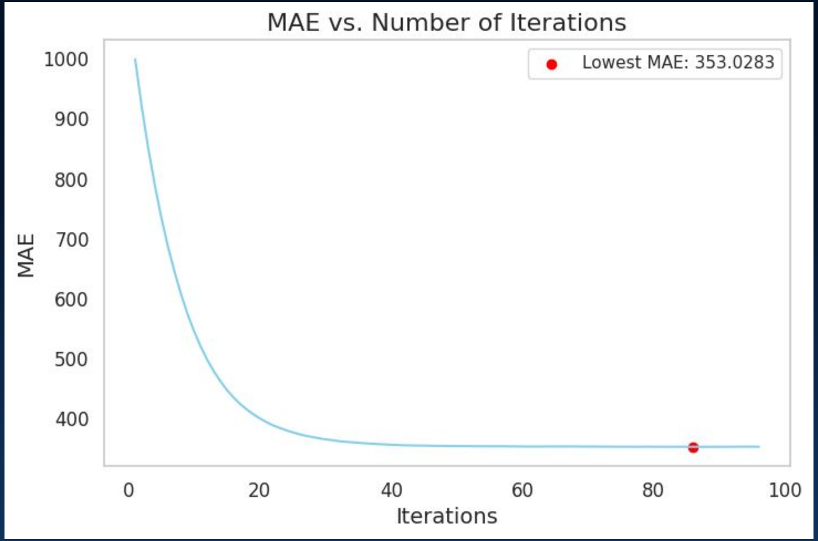
Final Hyperparameters	
# Estimators	200
Max Depth	10
Min Samples Leaf	4
Min Samples Split	10
Bootstrap	True

Gradient Boosting

Model

- XGBoost (eXtreme Gradient Boosting), is an ensemble learning method that uses the gradient boosting framework
- XGBoost builds decision trees sequentially, where each tree is trained to correct the errors made by previous trees

Final Hyperparameters	
Gamma	0
Learning Rate	0.1
Max Depth	4
Reg Lambda	1



Multilayer Perceptron (MLP) Neural Network

Model

- Feedforward neural network with at least 3 layers
- For supervised regression
- Used scikit-learn’s MLPRegressor

Normalization

- Normalized feature data using StandardScalar

Hyperparameters

- Tuned hyperparameters using grid search

Final Hyperparameters	
Hidden Layer Sizes	100, 50, 25
Activation	ReLU
Alpha	0.0001
Max Iterations	500

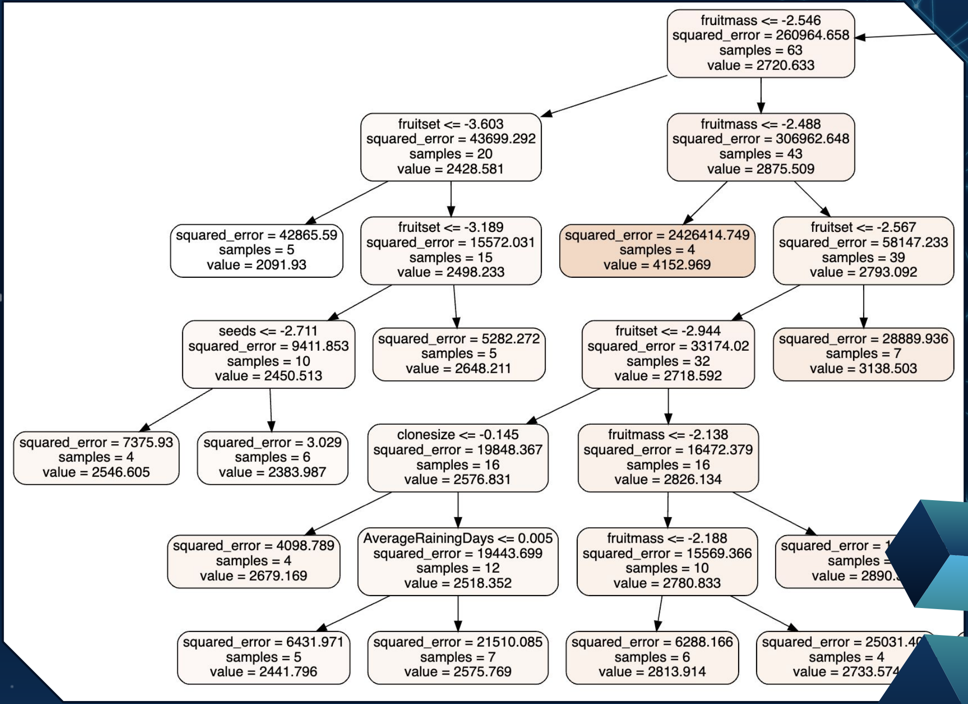
Metric Results

Model	MAE	MSE	MAPE	R^2 Score
Linear Regression	362.1	312,657.1	0.06375	0.8232
Ridge Regression	362.0	312,566.3	0.06378	0.8232
Random Forest	364.4	289,360.6	0.06069	0.8364
Gradient Boosting	353.0	324,612.6	0.06374	0.8188
MLP	366.7	301,962.3	0.06503	0.8292

Improvement Over Baseline

Model	MAE	Improvement?	Reason
Linear Regression	362.1	Baseline	
Ridge Regression	362.0	✓	Same feature and similar model, hence only a slight improvement
Random Forest	364.4	✓	Robust hyperparameter tuning helped achieve the best model for the dataset
Gradient Boosting	353.0	✓	Similar to random forest, a decision tree based model works well for the dataset
MLP	366.7	✗	Neural networks perform better for non-linear datasets

Best Solution Visualizations: Random Forest



Sample branch of a decision tree from the forest

References

[1] “Prediction of Wild Blueberry Yield | Kaggle,” Kaggle.com, 2024. https://www.kaggle.com/competitions/playground-series-s3e14/overview (accessed Jan. 30, 2024).

[2] K. P. Murphy, Probabilistic Machine Learning: An introduction. Cambridge: MIT Press, 2022. Accessed: Jan. 30, 2024. [Online]. Available: https://probml.github.io/pml-book/book1.html