Maize Flowering Time Estimation: Regression Analysis

Priyanka Gopi
Department of Computer Science
University of Central Florida,
Oviedo, Florida
priyankagopi@knights.ucf.edu

Abstract*— Predicting maize crop yields especially in maize production is paramount to alleviate poverty and contribute towards food security. Many regions experience food shortage because of uncertain climatic changes, poor irrigation facilities, reduction in soil fertility and traditional farming techniques. It is critical to breed new maize varieties and select for the desirable features, such as high yield and good stress tolerance, considering ongoing climate change and the decreasing quantity of arable land. Time-consuming and subject to human error, traditional phenotyping techniques rely on manual assessment. This study builds a regression model to estimate the days to anthesis male flowering time. Some of the features include Geno codes that includes Family ID and line ID, marker data and DtoA (response variable). Data on maize was studied with the goal of predicting the time at which a maize crop undergoes flowering - a regression task. This study was able to get 0.77 R2 value. The dataset, maize.sas7bdat is originally sourced from Buckler et al. (2009), "The Genetic Architecture of Maize Flowering Time," Science 325, 714-718 and coded in python 3.

Keywords—Linear regression, maize crop, regression analysis, feature selection

I. INTRODUCTION

The world is facing crop production challenges due to the rapid expansion of the human population and the adverse agricultural environments caused by climate change, urbanization, soil degradation, water shortages, and pollution. Breeding new varieties of major crops to achieve higher yield potential and stress tolerance is a promising solution to ensure food security. Maize is one of the most consumed grains globally, serving as an essential resource for human food, animal feed, and bioenergy. In 2020, the U.S. produced more than 360 million tons of maize, accounting for 34.28% of the total production worldwide. High-throughput phenotyping offers a great deal of promise to accelerate the breeding cycle's selection process and boost productivity. The main components of traditional phenotyping techniques are manual evaluations and visual ratings, which demand heavy workloads and are subject to human bias. As a result, it is crucial to create a cutting-edge high-throughput phenotyping (HTP) framework for breeding maize and other species.

Field trials are an expensive part of the breeding process even though the harvest is done using combines since human labor and time are needed to evaluate thousands of hybrid lines. To completely comprehend the physiological mechanisms underlying the hybrids with varied genetic backgrounds and their interactions with the environment, breeders frequently require thorough in-season data. Breeders can use these in-season measurement data to do genome-wide association analysis, analyze the genetics behind seedling growth and crop stress tolerance, and ultimately increase crop production. However, it takes a lot of time and effort to manually measure plant features in the field under growing conditions.

To maximize yield and seed quality in maize, flowering time is crucial when deploying cultivars in various conditions. By adjusting the vegetative and reproductive phases to local climate variables, the timing of flowering reveals how well a plant has adapted to its surroundings. It consequently correlates with aspects of its growth, such as plant height, the total number of leaves, and grain fullness. Drought, for example, will alter the maize blooming pathways and lengthen the anthesis-silking interval, which will have a negative impact on the rate of fertilization, kernel filling, seed quality, and weight. Therefore, in maize breeding efforts, keeping an eye on flowering time is essential for choosing superior genotypes for specific target settings.

Two statistical models are being used here which are linear regression model, Random Forest and Ridge regression model. In this project we intend to predict the flowering time of the maize crop using the Ordinary Least Squares, Linear Regression technique.

We used Python, an object-oriented language, for this project. Numerous libraries, including pandas, scikit-learn, NumPy, matplotlib, seaborn, etc., are available for Python. The panda's library, which is needed to retrieve the dataset and obtain the necessary metadata, was imported.

II. METHODOLOGY

The prediction algorithm developed may not provide the best predictions or predict actual flowering time but can help us narrow down values closer to them.

Steps involved in Data Mining:

1. Data Preparation

Being one of the crucial steps in data analysis, data preparation here involved are removal of unnecessary columns and converting the columns to float data type. The values containing both Numeric and string data were made to float data type to perform modelling. We also dropped duplicate variables as it could mislead the data patterns.

2. Data Exploration

Exploratory data analysis was conducted to visualize and understand the data given.

3. Filter out irrelevant features

Initially the data shape was (4981, 7393). Steps like dropping features that varied 5% of less across all rows using Variance Threshold and duplicated features. Variance of variables was obtained and the features with lower variance were eliminated as they can be considered as irrelevant due to the nearly similar values that they produce. Variables with low correlation to the response variable were eliminated too.

III. Analysis Of The Data

Some of the inferences from visualizations are:

Data is not skewed.

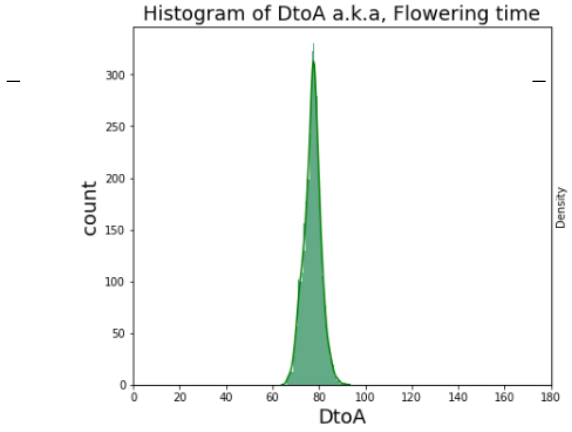

Fig. 1. Histogram of DtoA, Flowering time of Maize

The variation of DtoA variable for every Geno_code in the dataset is displayed below.

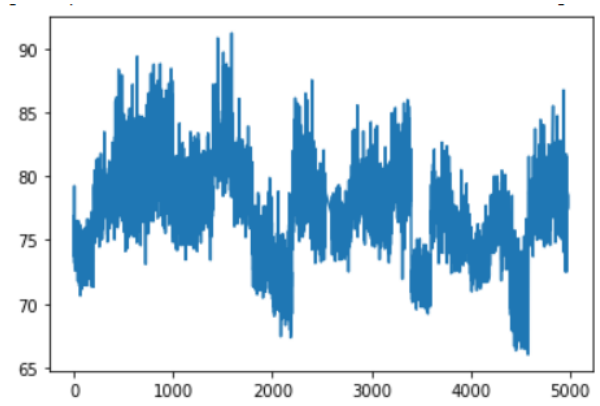

Fig. 2. Plot for x, DtoA (dependent variable) vs. y, Geno Code (independent variable)

Following splitting data for modelling, we made sure to normalize the data to make it ready for the modelling. The snippet of the code is below.

```
#normalize the data
scaler=StandardScaler()
scaled_X_train=scaler.fit_transform(X_train)
scaled_X_test=scaler.transform(X_test)
print(type(scaled_X_train))
print(scaled_X_train.shape)
```

```
<class 'numpy.ndarray'>
(3145, 7392)
```

Fig. 3. Code snapshot of data normalization

IV. Data Modelling

A. Linear Regression Model

The simplest form of regression is the linear regression, which assumes that the predictors have a linear relationship with the target variable. The input variables are assumed to have a Gaussian distribution. Another assumption is that the predictors are not highly correlated with each other (a problem called multi-collinearity). These models can be fit with numerous approaches. The most common is *least squares*, where we minimize the mean square error between the predicted values y^=Xβ^ and actual values.

$$y: (y - \mathbf{X}\beta)^2.$$

Here, the R2 score obtained is 0.06. The evaluation metrics for each one was the R2 score, and the root mean squared error (RMSE) values. The model is rigid and cannot be considered a good R2 value, hence a bad fit.

```
4880    78.1895
1713    79.0906
2148    73.1738
4452    66.7834
1578    84.1332
3455    70.4747
3036    80.3699
2735    74.7629
3580    73.4012
2111    72.0632
Name: DtoA, dtype: float64
[76.63131163 78.7907047  78.44812033 75.20728096 78.3596692  76.98621521
 79.29696232 75.4369494  77.82539895 79.66706265]
The intercept : 77.12064464228935

The co-efficients are:
[ 4.57041471e-02  1.94062339e-02 -3.22083365e-03  2.99851816e-02
  5.62867775e-03 -9.32697002e-04  4.74363607e-03 -5.39473860e-03
  1.72783325e-03 -1.64302104e-03  1.07728690e-02 -7.24453768e-03
 -1.72261070e-02 -5.22376763e-03  3.22501156e-02  6.73488775e-03
  1.32618680e-02 -1.17972110e-02  1.66694024e-02  4.72239482e-03
  1.00139545e-03  1.82449178e-02 -7.17036183e-03  1.44331448e-02
 -3.39332871e-03 -1.11052692e-02 -2.00909705e-02  1.74949128e-03
 -1.37044350e-02 -2.89021267e-03 -3.62908490e-03 -7.76364644e-03
  2.88616828e-02 -2.21819881e-02  8.26984191e-05 -1.57822907e-02
  2.37385366e-02 -1.96864736e-02 -2.05004682e-02  1.30785946e-02
  1.45039621e-02  1.74808670e-02  2.79921614e-02  9.17585116e-04
 -7.84323083e-03  9.12604486e-03 -2.32093068e-02  6.39364789e-03
  2.17666023e-03  1.30223848e-02  3.75496138e-03  2.26347376e-02
 -3.04689401e-02  3.69832226e-03 -9.37448085e-03 -2.51955653e-02
  2.48468655e-02  3.30969704e-02 -1.72374763e-02 -1.51956161e-03
  1.40150002e-02  2.43004225e-02 -3.02212346e-02  2.78706157e-02
  4.43114089e-02 -5.80342400e-04  1.90045978e-02  4.00455774e-02
 -1.50055730e-02  1.86182591e-02  1.90571743e-02 -1.14578801e-02
 -6.92197897e-03  1.53874739e-02 -2.10932078e-02 -4.67109437e-02
 -1.64536766e-02  2.41306343e-03 -2.66291086e-03 -1.41625846e-02
 -4.67815753e-02  5.05713792e-02 -3.84048773e-04 -2.39046168e-02
 -2.05338586e-02  1.79783047e-02  1.43902098e-02  3.18828935e-03
 -2.53248919e-02  1.56227532e-02  1.28224487e-02  2.15821019e-02
  2.61046193e-03 -3.73682127e-02 -1.03143343e-02 -1.04222339e-02
  2.31556851e-02 -3.63454932e-02 -2.92769340e-02 -1.53642491e-02
 -1.31923544e-02  3.08738766e-02  6.36262784e-03  3.28265584e-03
  2.07771666e-02  4.90874334e-02 -2.23135172e-02  2.05627571e-02
  2.13850611e-02  6.67108880e-02 -7.22170508e-03  1.86729478e-02
 -4.08341824e-02 -5.00951925e-02  5.41201575e-02  7.15322807e-03
  2.31813375e-02 -3.36512993e-02  1.54864296e-02 -1.76166927e-02
  1.18975440e-02  6.11409948e-02  2.13502429e-02  2.65192031e-02
 -2.33914744e-02  2.21590789e-02 -8.69928176e-02 -1.59374296e-02
 -1.70530538e-02  2.65691899e-03  7.23972149e-02  4.27698085e-02
  1.15099806e-02  3.17921538e-02  9.83472553e-02  1.02874969e-01
  1.52233073e-02  4.01875955e-03 -4.85836844e-02  1.57669887e-01
 -6.92956306e-02  3.72033125e-02  1.55934470e-02  3.83558609e-02
 -2.20758354e-01  2.02430265e-03  5.46650761e-03]
The RMSE is : 3.5033730049069947
The R2 score: 0.061765133192504096
```

Fig. 4. Code snapshot of linear regression function parameters

B. Ridge Regression Model

Ridge regression is an extension of linear regression where the loss function is modified to minimize the complexity of the model. This modification is done by adding a penalty parameter that is equivalent to the square of the magnitude of the coefficients.

Loss function = OLS + alpha * summation (squared coefficient values)

In the above loss function, alpha is the parameter we need to select. A low alpha value can lead to over-fitting, whereas a high alpha value can lead to under-fitting. In our case, Ridge regression didn't work well probably because of many variables, which can also considered to be another drawback of ridge regression.

```
8.149665256888378
-4.077130789388804
```

Fig. 5. Code snapshot of Ridge function parameters RMSE and R2

C. Random Forest Model

The Random Forest model worked much better than the linear regression, in our case. We managed to obtain the R2 value of 0.77 which is the best performance of the model for the features count of 4000 and sample size of about 4500. This also shows how large features can work the best with large sample size.

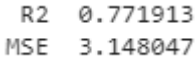

Fig. 6. S napshot of Random Forest model parameters RMSE and R2

D. Feature Selection

The model gave the best R2 score for best 4000 features in this case. We used the 'f_regression' which is a correlation calculator and obtained the best features from the same.

Conclusions

The important features is the Geno_code, pop along with the marker data columns (m434, m435, m436..more) which indicates that these parameters play a huge role in predicting response correctly. We got a R2 score of 0.77 with a RMSE value of 3.14, which can be considered as a decent fit.

Acknowledgement

I thank Dr. Rui Xie for the support and incorporating this project in the course syllabus.

References

[1] Waldmann, Patrik, Gábor Mészáros, Birgit Gredler, Christian Fuerst, and Johann Sölkner. "Evaluation of the lasso and the elastic net in genome-wide association studies." Frontiers in genetics 4 (2013): 270.