Machine Learning
Agriculture Yield Prediction

by Data Pirates:
Kashish Shah-AU1841014
Harsh Patel-AU1841015
Hriday Nagrani AU1841042
Yugamsinh Chavda AU1841090

***Abstract*—This document focuses on building a predictor which predicts the agriculture yield using the Ridge regression model.The agricultural yield of any crop depends on different Argo-Climatic factors like rainfall, temperature, soil type and the area and production of that crop.**

***Index Terms*—Supervised Data, Crop Yield,Regression, Data Augmentation, Data Normalization, learning rate, Gradient descent, Regularisation, hyper-parameters.**

I. Introduction / motivation, and background

Agriculture sector is one of the most important sector in Indian economy. With the continuing expansion of the human population and urbanization, understanding agricultural yield is central to addressing food security challenges.

Studying the impacts of climatic change on crop yield has become one of the most important topics to look forward as our surroundings are changing due to the impact of global warming.

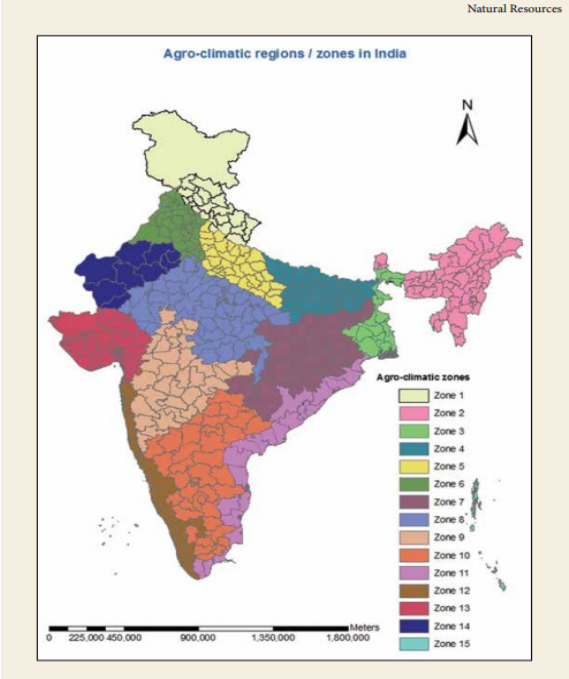
Better accuracy while predicting the yield is desirable because an accurate prediction helps the farmers to decide on what to grow, when to grow and how much to grow.

II. Literature Survey

We visited many official government websites like :
1) https://iasri.icar.gov.in/
2) https://agricoop.nic.in/en
We explored through these websites and read about the agricutural scenario of india. We also went over the latest official Government agricutural report and also the estimate prepared by the government.

Here are the main learning outcomes :

1) Agro-climatic regions/zones in India : India is divided into 15 major agro climatic regions with multiple parts of state covering each region.



2) Agro-ecological regions in India:
   Arid
   Semi Arid
   Sub Humid
   Humid-PerHumid
   Coastal
   Island
3) Broad soil groups:
4) Major Soil categories:
5) State and zone wise fertilizer and pesticides conumption:
6) Production yeild and growth of major crops like rice and wheat.
7) Export and import trends and the position of india in world agricutural frameworks
8) Estimated and targeted yield sheet for 2020-21 by Ministry of Agriculture and Farmers Welfare.

III. Implementation

A. Data Preprocessing

After collecting data for different factors such as rainfall, soil type, fertilizers, there were many outliers in the data, missing values, Variable transformation and many other issues.

We removed the outliers by replacing the values with the mean value. Missing values are replaced with mean value. In

soil data we have different components of soil in a single cell. We divided the soil data individually and then created new columns and filled the values.
We transformed multiple columns to a single column by taking average of them or just by adding them. For example, in the fertilizer data, we had two columns for Nitrogen fertilizer(variable), one for kharif season and other for rabi season but as we are considering annual data, we added them into a single column for Nitrogen fertilizer.

B. Implementing Ridge Regression from scratch

Ridge regression is like an upgraded version of linear regression. In linear regression we minimize the residual sum of squares(RSS or cost function).
The resulting cost function is :

$$L(m)=\frac{1}{m}\times\sum_{i=1}^{m}(y^i - h(x^i))^2$$

But the problem with the linear regressor is it consider all the features equally weighted while minimizing the cost function. No bias is introduced and the variance will be high. This results in overfitting. Means, the model will perform well on seen data but poor on unseen data.
Now, ridge regression can deal with this problem. In ridge regression a l2 penalty term is introduce, in the cost function. The penalty term is sum of square of magnitude of the weights. This will basically penalize the weights which are contributing towards the high variance and hence the overfitting will be reduced.
Modified cost function for ridge regression is:

$$Penalty=\lambda\sum_{j=1}^{m}(w_j^2)$$

$$L(m)=\frac{1}{m}\times\sum_{i=1}^{m}(y^i - h(x^i))^2 + \lambda\sum_{j=1}^{m}(w_j^2)$$

While applying gradient descent, the l2 penalty term will penalize the weights of the model and will make our hypothesis simpler. This will introduce generalization in the model because it will reduce overfitting. We tuned the following hyper parameters for getting the best results:

1) Learning rate
2) l2penalty
3) number of iterations

C. Implementation of Gradient Boosting(using sklearn) for comparison

Gradient Boosting uses decision trees as a weak learner for prediction. The decision tree creates a low bias and high variance scenario but the gradient boosting reduces the variance and hence it takes us out from the overfitting situation. We tuned the following hyper parameters for getting the best results:

1) Learning rate
2) max depth
3) number_estimators: no of boosts given

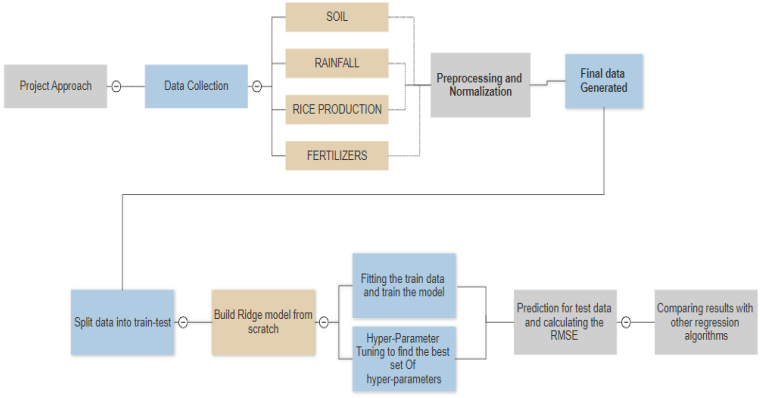
D. Implementation of Random forest(using sklearn) for comparison

In random forest, random samples of rows and features(row sampling and feature sampling) are taken and passed as an input to the decision tree. The number of times this process is done depends on the hyper parameter 'number_estimators' because that much trees we are taking. After all the decision trees gets trained, in the prediction stage, the average of the outputs of all the trees is taken as the final prediction.

We tuned the following hyper parameters for getting the best results:

1) max depth
2) number of estimators
3) random_state

IV. Flow of the Project

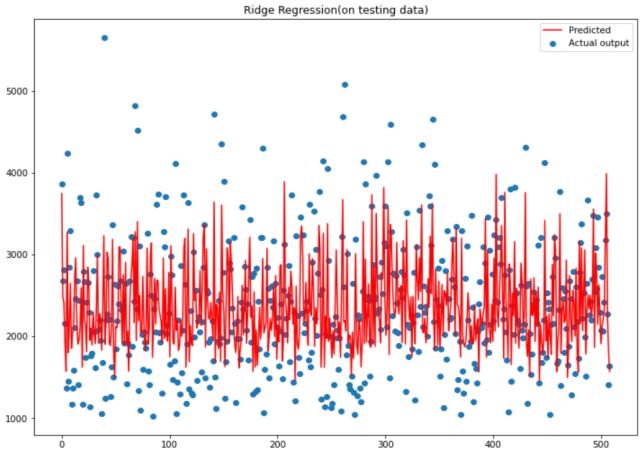


V. Results

A. Preprocessed Data information

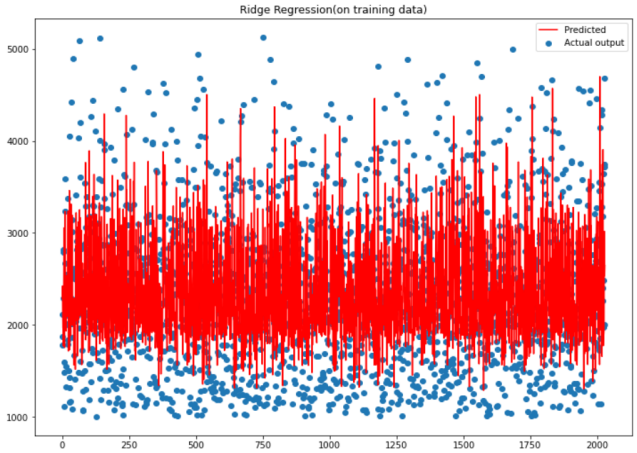
```
Data columns (total 27 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Dist_Code        3210 non-null   int64
 1   Year             3210 non-null   int64
 2   State_Code       3210 non-null   int64
 3   State_name       3210 non-null   object
 4   Dist_name        3210 non-null   object
 5   ANNUAL           3210 non-null   float64
 6   avg_rain         3210 non-null   float64
 7   Nitrogen         3210 non-null   int64
 8   POTASH           3210 non-null   int64
 9   PHOSPHATE        3210 non-null   int64
 10  DYSTROPEPTS      3210 non-null   float64
 11  FLUVENTS         3210 non-null   float64
 12  INCEPTISOLS      3210 non-null   float64
 13  LOAMY_ALFISOL    3210 non-null   float64
 14  ORTHENTS         3210 non-null   float64
 15  ORTHIDS          3210 non-null   float64
 16  PSAMMENTS        3210 non-null   float64
 17  SANDY_ALFISOL    3210 non-null   float64
 18  UDALFS           3210 non-null   float64
 19  UDOLLS_UDALFS    3210 non-null   float64
 20  UDUPTS_UDALFS    3210 non-null   float64
 21  USTALF_USTOLLS   3210 non-null   float64
 22  USTALFS          3210 non-null   float64
 23  VERTIC_SOILS     3210 non-null   float64
 24  VERTISOLS        3210 non-null   float64
 25  RICE_PRODUCTION  3210 non-null   float64
 26  RICE_YIELD       3210 non-null   float64
dtypes: float64(19), int64(6), object(2)
memory usage: 702.2+ KB
```

B. Ridge regression on testing data



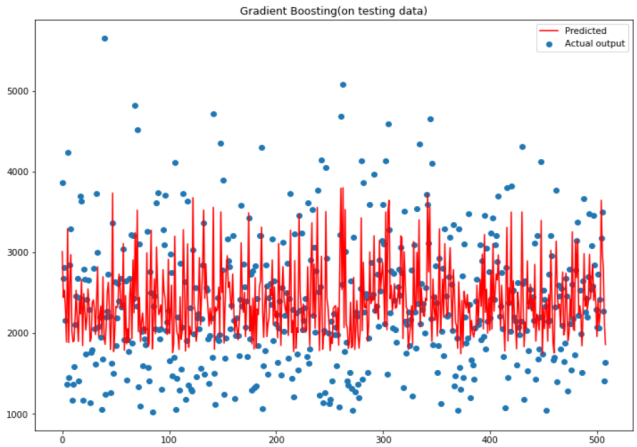
mean_squared_error = 579.5211993555529

C. Ridge regression on training data



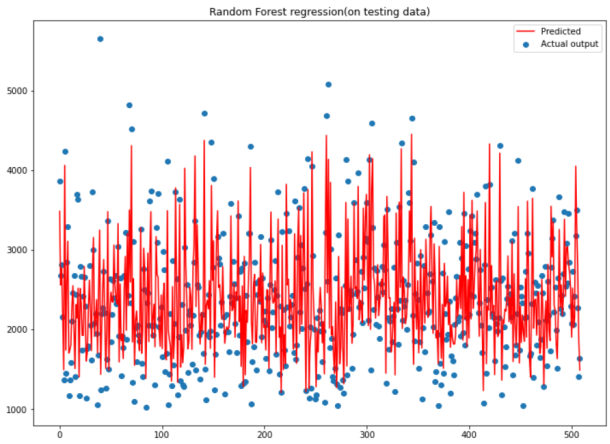
mean_squared_error = 607.6989792488598

D. Gradient boosting regression results



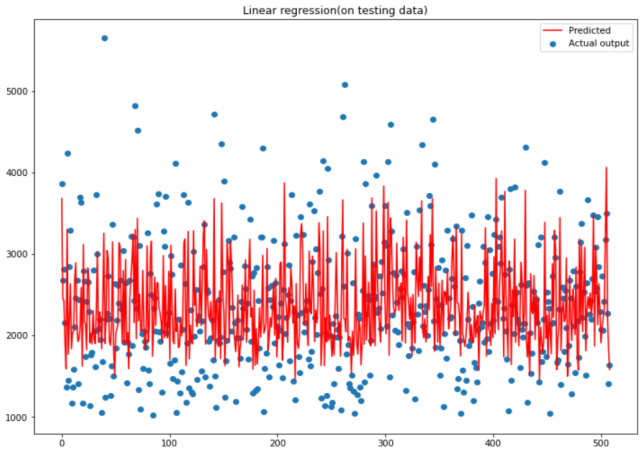
mean_squared_error = 520.7190620379137

E. Random forest regression results



mean_squared_error = 415.06975729921913

F. linear Regression results



mean_squared_error = 581.3990624951123

VI. Conclusion

After applying four different regression methodologies which are: Linear Regression, Ridge regression,Random Forest and Gradient Boosting with best possible set of hyper-parameters, we found out that the Random Forest has the best results as root Mean squared error for this comes out be minimum. Not a drastic change in rmse is observed on applying ridge regression as compared to the linear regression. Also, Gradient boosting technique performed better than the linear regression and the ridge regression techniques but not as good as Random forest regression techniques.

References

[1] https://iasri.icar.gov.in/
[2] http://data.icrisat.org/dld/src/crops.html
[3] https://data.gov.in/
[4] https://scikit-learn.org/stable/modules/linear_model.html
[5] https://www.youtube.com/watch?v=Nol1hVtLOSg
[6] https://www.youtube.com/watch?v=nxFG5xdpDto
[7] https://tn.data.gov.in/catalog/statistical-hand-book-2019-agricultureweb_catalog_tabs_block_10
[8] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
[9] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html