Data Science Challenge Email: vivek8943@gmail.com
Mobile:+1806-939-5833

Data Challenge

Problem:

- We have a time series dataset with a continuous target variable – Wheat Yield. The aim of the predictive modeling is to build a model that can predict the yield by learning from the given training data and predict the yield on unseen test set. The model should be generalizable to be used on unseen real world test set.

- As the target variable is continuous we have to use regression techniques to predict the Yield based on the features provided. As the large data is available for training, there is a high possibility of over-fitting and suitable methods like L1 and L2 regularization has to be use to reduce the over-fitting and to improve the generalization of the learning algorithm.

Out Line of solution:

1.Exploratory Data Analysis
- Correlation of variables
- Distributions
- Outliers detection
- Visualization Hierarchical Clusters
-

2.Data Pre Processing
- Scaling the Values
- Missing Value
- Feature Selection and Engineering

3.Predictive Modeling
- Linear Models
- Random Forest (Bagging)
- Gradient Boosting
- Stacking (Meta Learners and Base Learners)
- Grid Search for Hyper Parameter tuning

4.Calculating RMSE for predictions

Data Science Challenge Email: vivek8943@gmail.com
Mobile:+1806-939-5833

## Understanding the data

*Questions on Assumptions:*

1. Neglecting Date from the data set:

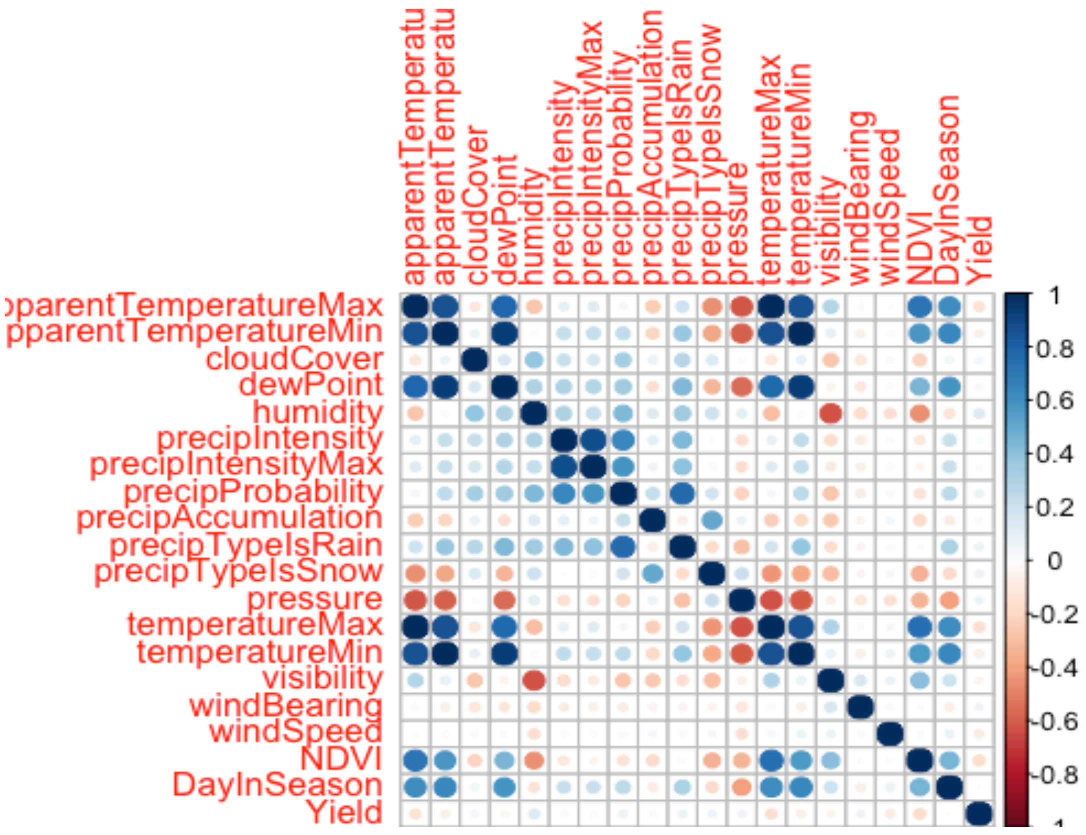
We have an inherent seasonality and trend that can be extracted from the time series data. Neglecting the Date would deprive us of this information although day in season could capture a little of seasonality but misses cyclic trends.

2. Neglecting County and State

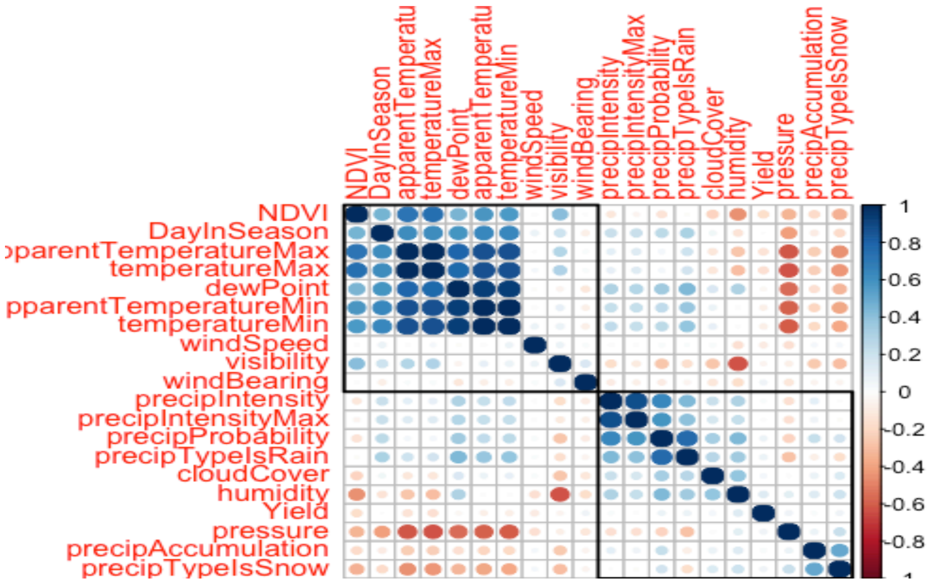
Every county/State might have certain factors that affect the crop yield that are latent or other than the features we have collected. Ideally there should be a separate model for different categories with different trends of yields (avoiding Simpsons Paradox). We could use the County and State and use regularization (L1 and L2 norms) to reduce the over-fitting.

## Exploratory Data analysis:

**Correlation Plot:** The correlation plot show that there are no highly correlated variables and Temp Max and Temp Min as the Min temperature is usually a dependent on Max temperature and vice versa.



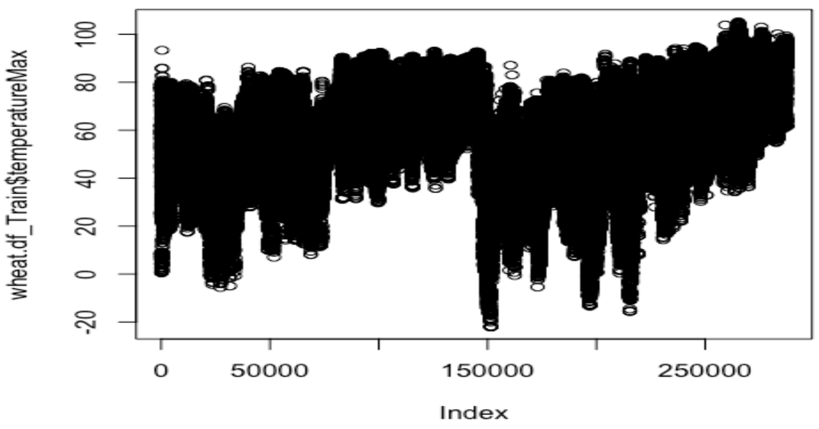
Hierarchical Clusters Present in the data:



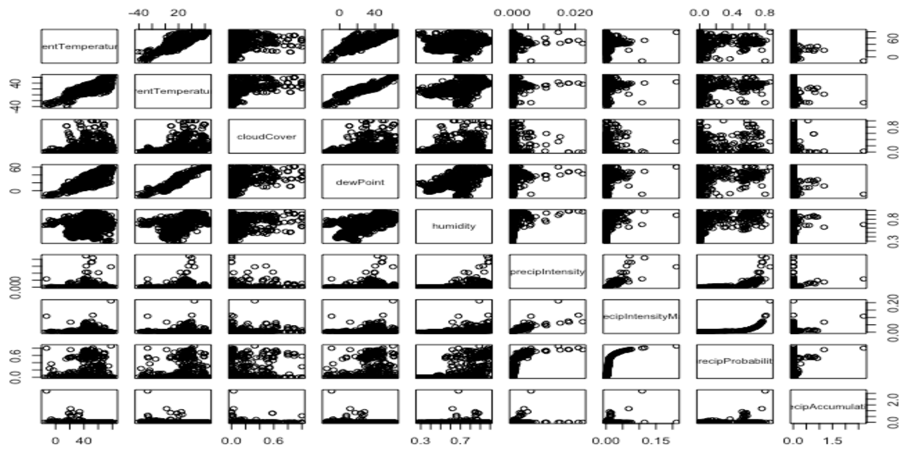
From the diagram we can see that there are two clusters with related features that can shows that they are related.

Email: vivek8943@gmail.com
Mobile:+1806-939-5833

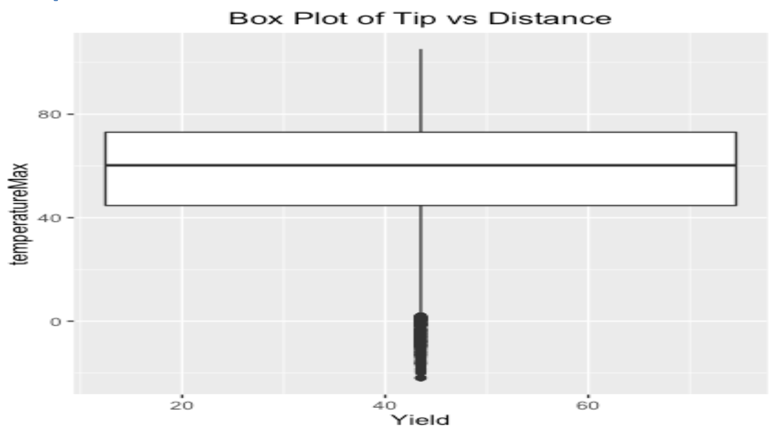
## Distribution of temperature



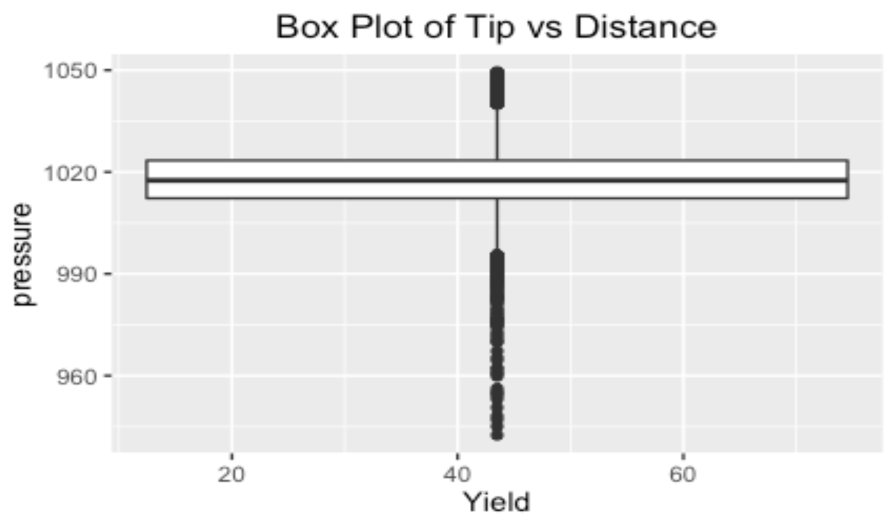
**Multivariate Scatter plot**: There is a Linear trend in most of the variates and few are skewed and need transformation when used with linear models.



## Box plots:



Email: vivek8943@gmail.com
Mobile:+1806-939-5833



Although these look like outliers, these should not be removed as they donot seem to be by chance but variations in temperature and pressure.

## Data Pre-Processing

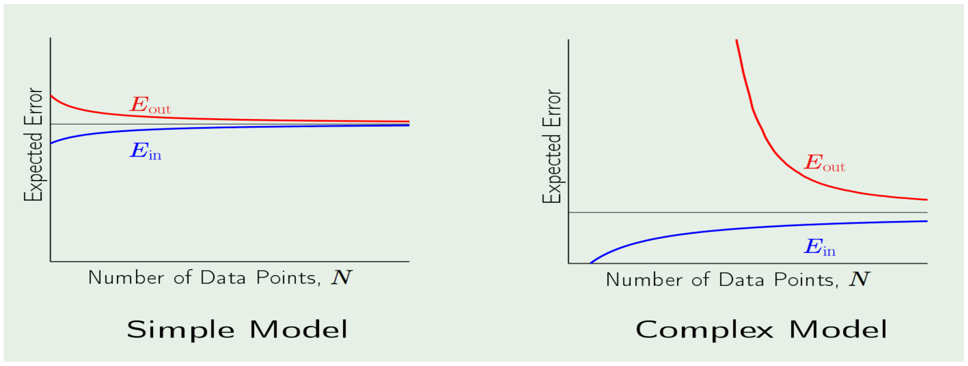
- Split Dataset into Test set (20%) and Training set (80%) and K fold cross validation is used in the predictive models
- Normalizing and scaling of data, as the range of values of different features is different.
- Check for missing values and formulate a way to impute the missing values. Missing value percentage is very less so only complete cases are considered
- Feature Selection: The random forest algorithm predicts the variable importance as it calculates the Gini coefficient while splitting the trees. So the estimate from the below random forest shows that almost all features play an important role in prediction except the constant column *precipTypeIsOther* that has been removed from the data set.
- Feature Engineering is based on the domain knowledge and few new features can be generated from the feature set to account for the missing 15% of the R2 value.

## Using Learning algorithms

- Advantage of using non-Linear methods is, the data need not strictly follow the assumptions of the Linear Regression namely: 1)Linearly dependency 2)Homoscedacity 3)Multi collinearity 4)Auto correlation.

- The complex models can reduce the Out of Bag Error (RMSE), when the number of examples in the training set is large.

Email: vivek8943@gmail.com
Mobile:+1806-939-5833



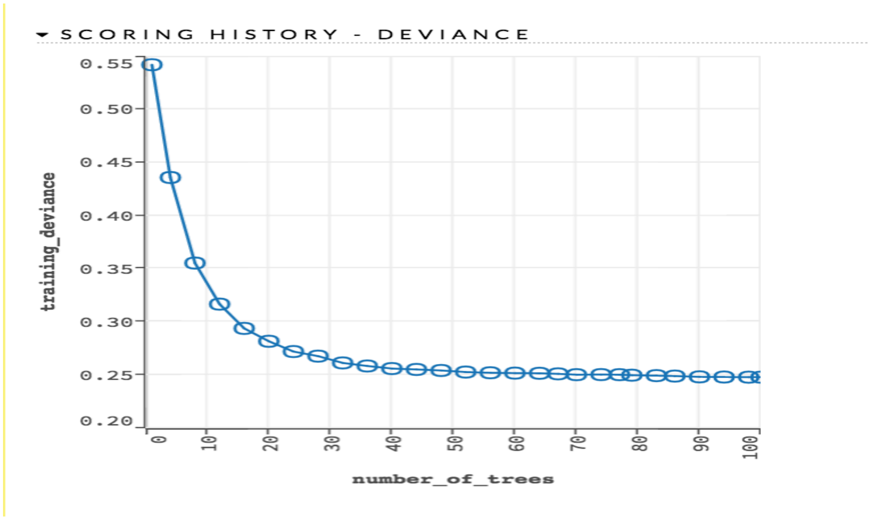
- As we have large number of data points we can afford to consider complex non-linear models by penalizing it for the VC dimension to reduce the generalization error.

## Measures to prevent Overfitting:

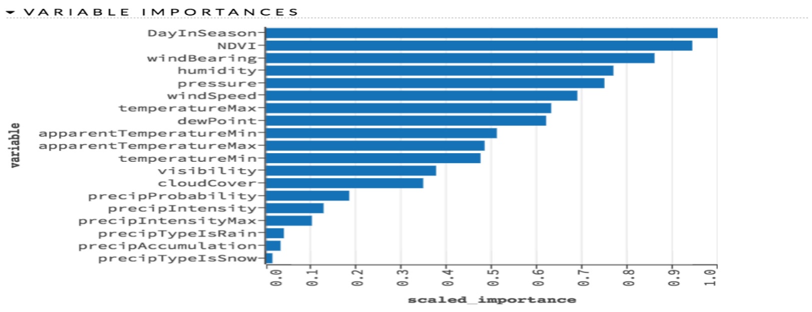
1.Early Stopping
2.K-Fold Cross Validation
3.Regularization (l1,l2 norms)

## Random Forest Without Cross Validation:

Below are the plots for the Training deviance and variable importance.
Training Deviance:



## Variable Importance:

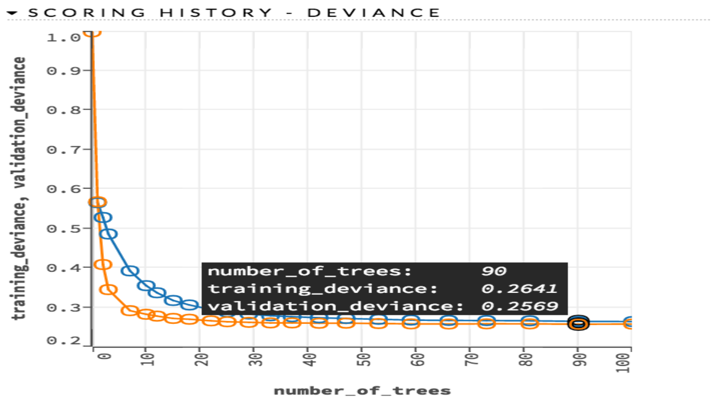


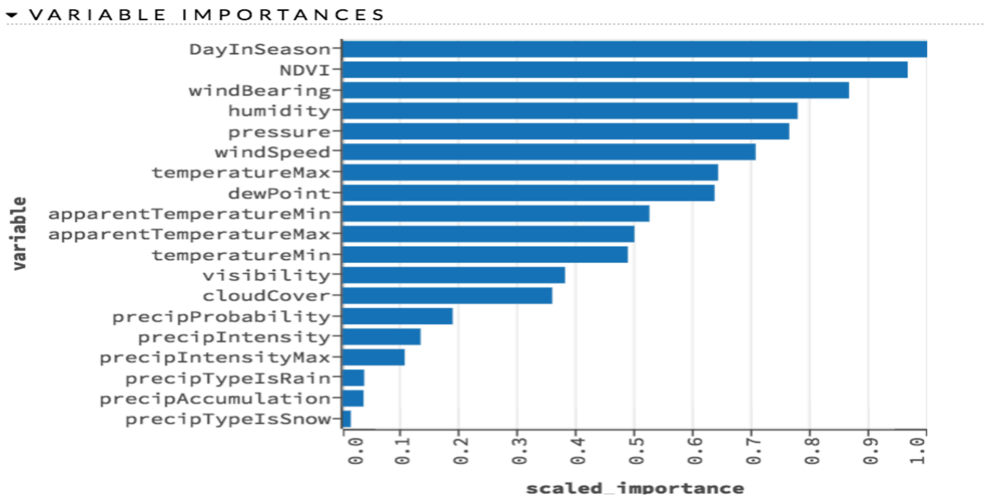
- Day in season is most important feature in the data set as it captures the seasonality aspect from the time series data.
- Most of the variables play major role in representing the variation of the target variable.

## Random Forest With 10- Fold Cross Validation:

The below curve suggests that the cross validation curve almost traces the training curve. So the generalization error of the hypothesis function h(x) generated the Random forest algorithm is less.

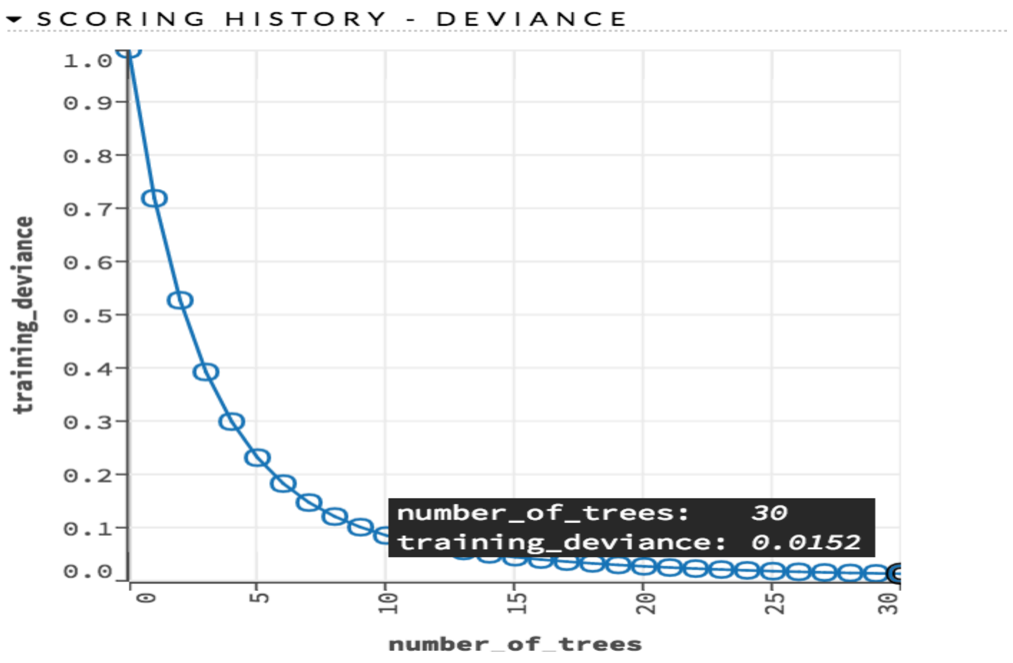
To further improve the generalization of the algorithm, we can reduce the number of trees to 30-40 as the reduction of error is not so significant.





## Gradient Boosting Machines:

In Gradient Boosting, by sequentially applying weak learners to the incrementally changed data, a series of decision trees are created that produce an ensemble of weak prediction models. The Stochastic Gradient of the cost function is computed to attain the least possible RMSE value.



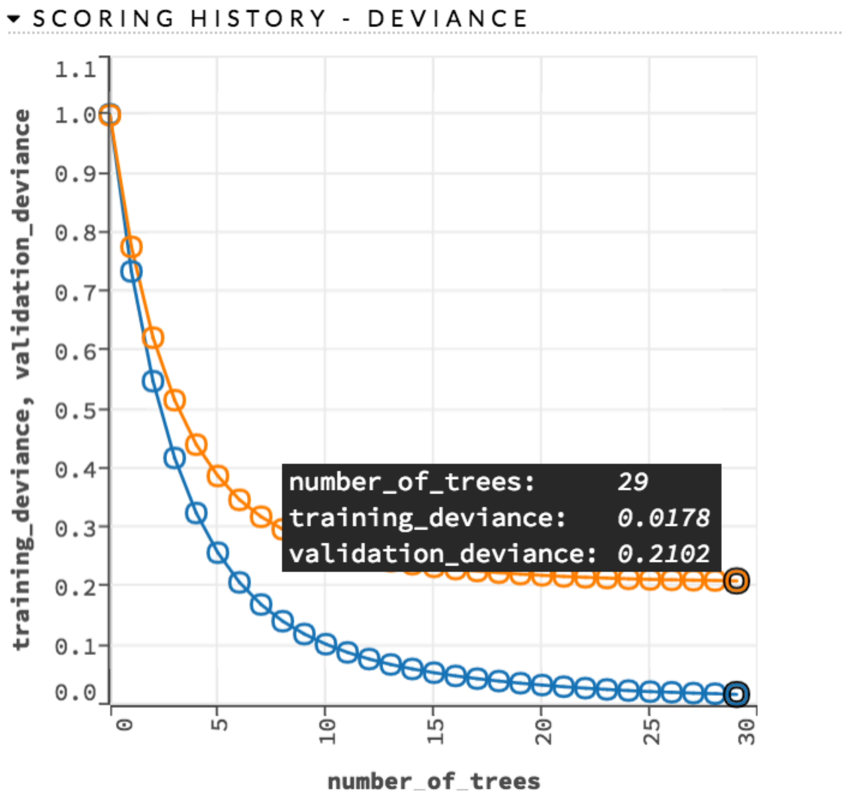
Email: vivek8943@gmail.com
Mobile:+1806-939-5833

## Cross Validation Deviance:

## Bias – Variance Trade Off:

A too complex model has low bias but large variance, while a too simple model has low variance but large bias, both leading a high error but two different reasons.

From the cross validation curve and training curves, it can be understood that the randomness introduced by the distributed random forest is reduced the variance but to reduce the bias in predicting the Tip Percentage we have to choose some other algorithm.
So it sounds a good idea to use Adaptive boosting or gradient boosting to reduce the Bias of the model



- Early stopping increases the generalization of the model by reducing the overfitting by not allowing the learning algorithm to learn the noise of the training set.
- The learning rate is chosen not in extremes as it might affect the convergence of cost function in a negative manner.

*Performance Metrics:*

*H2ORegressionMetrics: gbm*

*MSE: 0.1746179*
*R2 : 0.8253797*
*Mean Residual Deviance : 0.1746179*

*Out of Sample Error or the error on the Test Set:*

*MSE: 0.1776*
*RMSE: 0.421873*

- The GBM algorithm has an R2 value of 0.82 which means that 82% of the variation of target variable is explained by the Feature set used.

***Super Learning/Stacking model:***

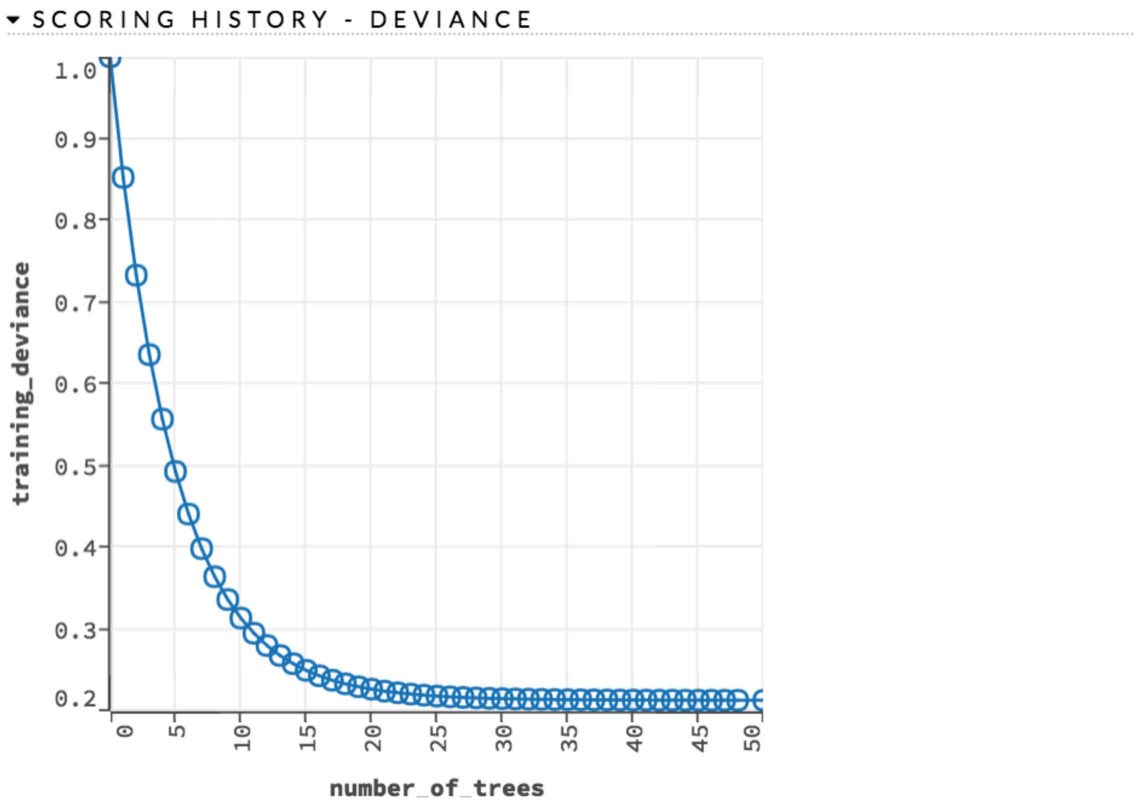
*Base Learners:* 1.GBM, 2.Random Forest, 3.Generalized Linear Model with 5-Fold cross validation.

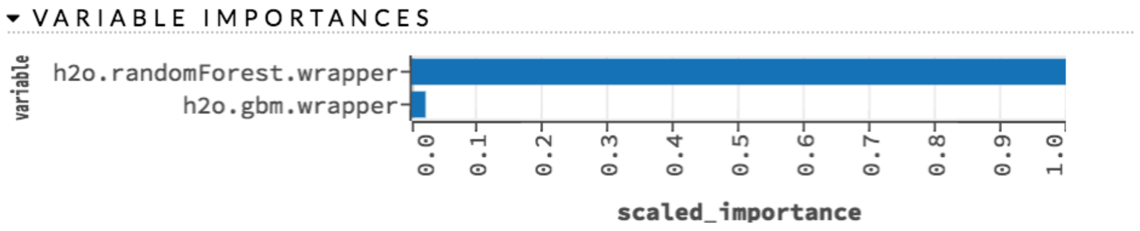
*Meta Learner:* Gradient Boosting Machine.

As an initial step, generic Base model wrappers are used to check the accuracy of predictions.

Below is the training deviance of the meta-learning algorithm (GBM), which scaled the importance of Random forest algorithm higher than the GBM. The GLM is almost given zero importance relative to other base learners so it is eliminated.

Email: vivek8943@gmail.com
Mobile:+1806-939-5833





OUTPUT:
RMSE : *0.401231*
R2: 0.8196

- We have a better RMSE value from stacking algorithm and 81% of the variance of the Target is explained by the predictive model.

Other Algorithms used:

- Linear Regression with L1 norm is not as effective as the above Ensemble methods (RF and GBM). Finding the right value for Lambda is important because it decides the shrinkage of the coefficients. Lasso regression induced bias into the model by shrinking the coefficients of few features to zero.

- Elastic Net Regression can be used as to overcome the limitations of the Lasso regression but we have to perform a grid search on the values of “Alpha” to generate better predictions.

Output for Regularized GLM:

```
       regularization Elastic Net (alpha = 0.5, lambda = 3.571E-5 )
        lambda_search nlambda = 100, lambda_max = 0.0, best_lambda =
                      3.571E-5
          predictions ·
                  MSE 0.893399
                   r2 0.106598
mean_residual_deviance 0.893399
```

The R2 value is very very less, so this model cannot be considered at all.

Artificial Neural Networks:

The Neural Network with the chosen hyper parameters give a very unstable performance and it requires hyper parameter tuning using grid search for which the code is written but it needs more time to converge as it generates large number of models with different combinations.

Although the RMSE is low the R2 value is less for the Neural Network algorithm so this requires more tweaking.

Mistakes and lessons learnt!

1.Data Snooping: The normalization of data should be done only after the Test and Training datasets are split. Otherwise the mean, min, max that are used in are calculated considering the feature vectors from the test set. So we are taking an unfair advantage by using some information from the test set.

Best Practices:

1. Always the predictive model should be simple but not simpler. Penalizing for the complexity of the models using regularization gives us a better predictive power than simpler linear models using few features.

Email: vivek8943@gmail.com
Mobile:+1806-939-5833

2. Try to use ensemble techniques to have a good bias – variance trade off. When variance of the model is high use bagging, when the bias is high use boosting techniques like gradient boosting and adaptive boosting

3.Further Work:

1.Feature Engineering – Use domain knowledge to create few features from the existing features
2. Grid search for hyper parameter tuning – I have skipped the parameter tuning for the learning algorithms to get the results faster as this is an exercise. Ideally the parameter tuning results in a better prediction with out overfit when used with early stopping and regularization.

- I would choose gradient boosting machine learning algorithm, which produced the best results. The generalization of GBM is much better which can be seen from the validation and training curves. The R2 value of the predictive model is about 82% with out feature engineering and Grid search for parameter tuning, which is pretty good.

- We could consider the Stacking method, but there is not much improvement from the results of GBM, so for a minor decrease in error the extra complexity of base learners and meta-learners can be avoided.

- If we have to explain the results of the prediction we have to carefully tweak the GLM or CART decision tree to get an interpretable model but that can reduce the predictive power of the models.

- I would strongly suggest using different models for different states/ regions as there are many latent factors that effect the yield in different regions and can result in better predictions.