Discovering Optimal Models For Real Estate Prices Prediction Using Linear & Non-Linear Regression Techniques

Areeb Waseem (18L-1823)

Rehman Shahid (18L-1890)

Samama Imtiaz Butt (18L-1882)

ABSTRACT

We know that real estate prices follow specific trends that depend on location, condition, features etc. This project aims at applying advanced **Regression Techniques** to predict the Real Estate Sale Prices given a set of features. The dataset will be a combination of several features such as interior, exterior, area, location, heating, number and condition of rooms etc. We will then train our models depending on these features and the respective sale prices. In the end we will run our model to predict house sale prices and test the accuracy of our results.

INTRODUCTION

- The goal of this project is to predict Real Estate Prices in the Houses sector.
- The Sale Prices follow certain tendencies of better condition leading to better selling price.
- Dataset is taken from an active Kaggle competition which consists of different characteristics of the houses in a csv file.
- Detailed analysis of the Dataset.

INTRODUCTION

- We will first Pre-process Dataset to make it appropriate for building our model.
- Then we will train our model by applying different Regression Algorithms.
- Next, we will test our model on the dataset and perform an analysis of the predictions of the prices.
- In the end, we will do a comparative analysis of different regression algorithms used in our model.

PROBLEM STATEMENT

The task is to build a model based on Machine Learning techniques that takes imperative features of the houses and predict their selling price. Our proposed solution will include different Machine Learning algorithms like ANN, Gradient Descent, Stochastic Gradient Descent, Random Forest Regression, Multi-variate Regression, Kernel Ridge Regression etc. and analysis based on their results. We will be implementing the solutions in raw python & also use libraries in some cases for building an efficient prediction model.

Dataset Analysis

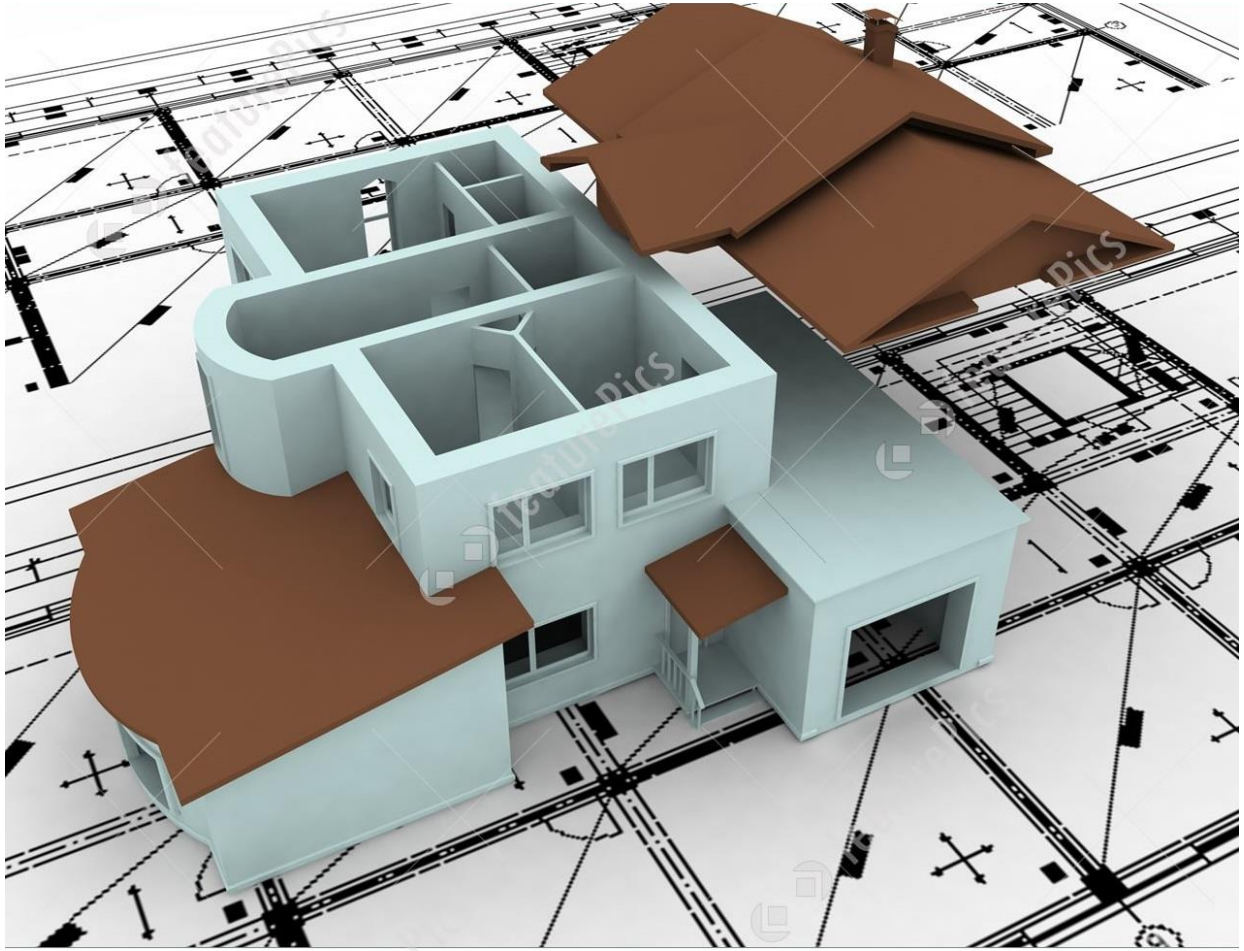
- Dataset consists of several features of the houses and respective prices. Data is strongly & linearly correlated which forms the basis for Machine Learning models.

The Sale Price shows following trends:

- *Deviate from the normal distribution*
- *Have appreciable positive skewness*
- *Sales Price and GrLivArea have a linear relationship*
- *Sale prices increase per year*

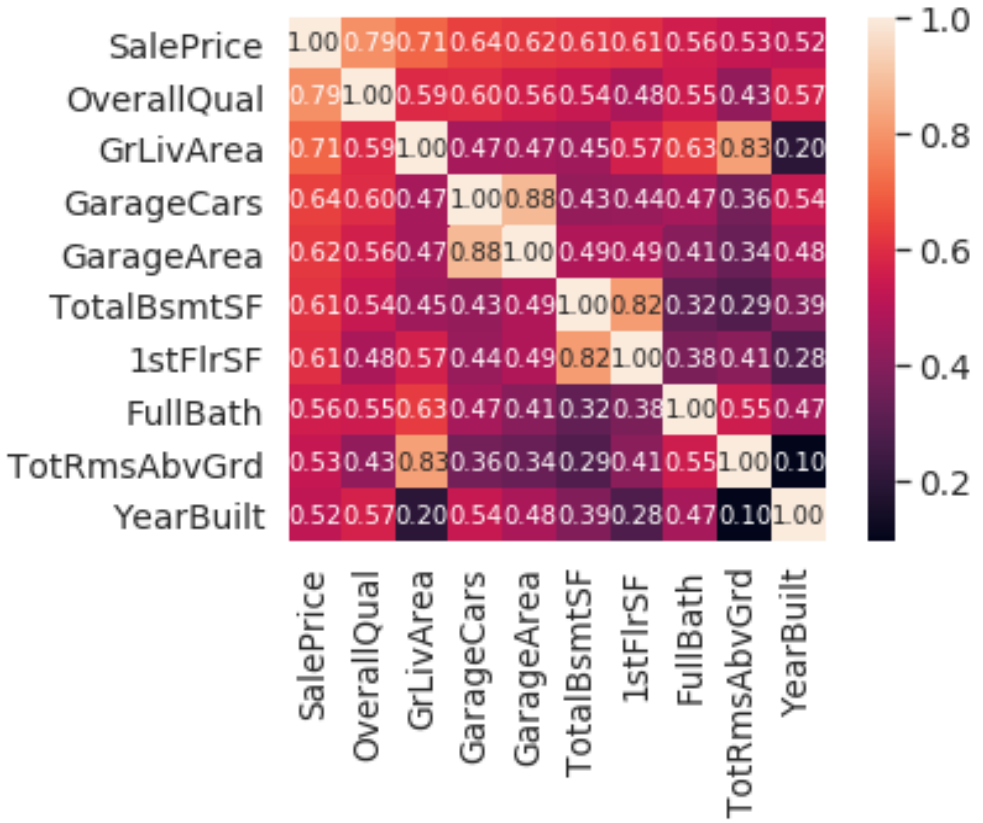
Deterministic Features for Price Prediction

- Sale Price
- Building Class
- Building Location
- Construction Type
- Condition / Quality
- Utilities / Access
- Year Built
- Square Feet
- *# of Floors*
- *# of Beds / Kitchens*
- *Roof Type*
- Garage Area



Highly Correlated Features

- **Sale Price**
- OverallQual
- GrLivArea
- GarageCars
- TotalBsmntSF
- 1stFlrSF
- FullBath
- TotRmsAbvGrd
- *YearBuilt*



Dataset Preprocessing

- Handling Missing Values & Type Conversions.
- Standardizing Numeric Data.
- Handling Skewed Data.
- Converting Categorical Data to Dummies.
- Split train & test features.

Dataset Preprocessing

Handling

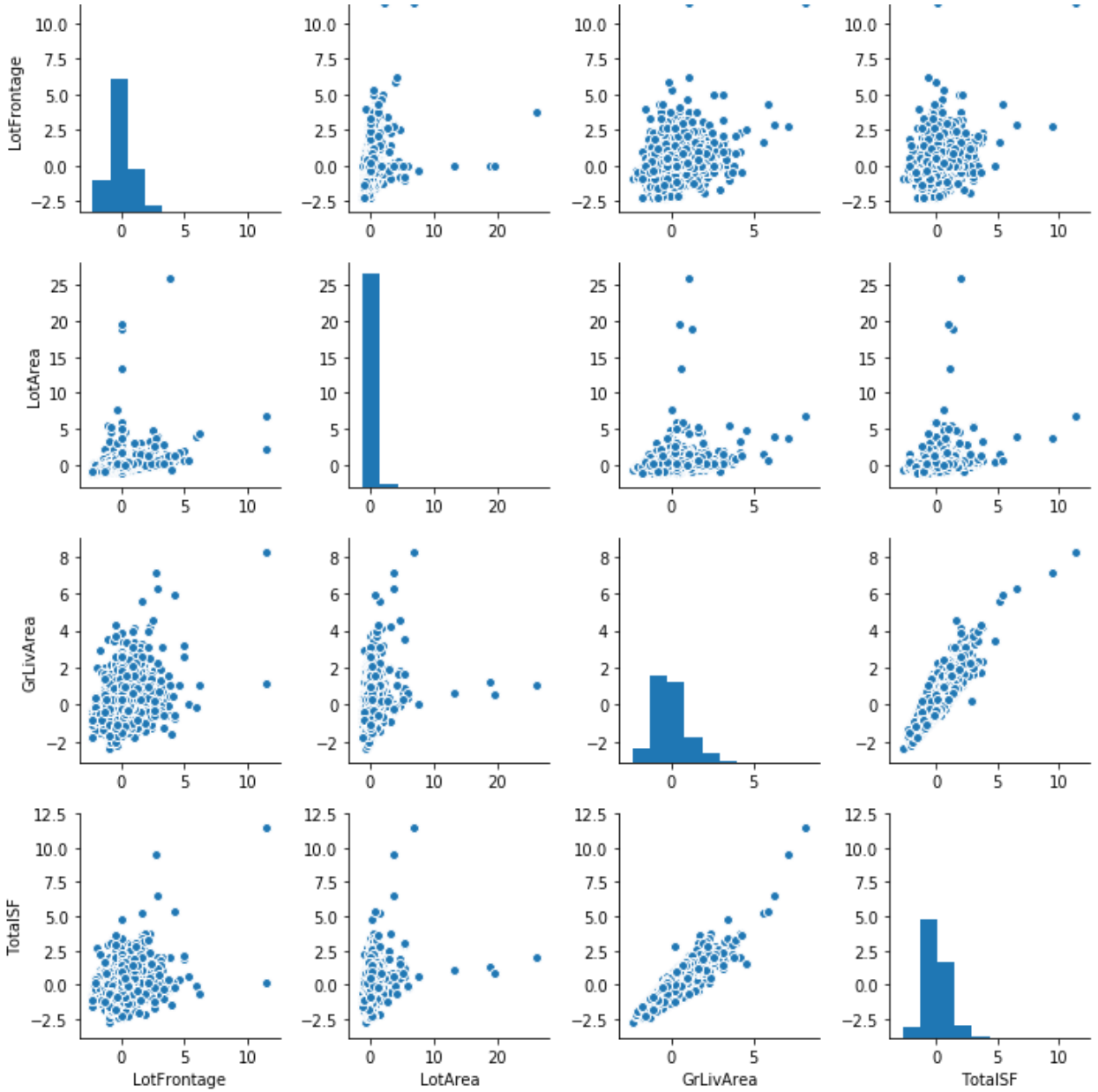
- Missing Values
- Type Conversions

Feature Name	Conversion
MsSubClass	toStr
MsZoning	Fill Na with most common feature (mode)
LotFrontage	Fill Na with Mean
Alley	Fill Na with No Access
OverAllCond	toStr
MasVnrType	Mode
Bsmt	Fill Na with No Basement
TotalBsmtSF	Fill Na 0
Electrical	Mode
KitchenAbvGr	toStr
KitchenQual	Mode
FireplaceQu	Fill Na with NoFp
Garage	Fill Na with NoGRG
SaleType	Mode
YrSold	toStr
MoSold	toStr

Dataset Preprocessing

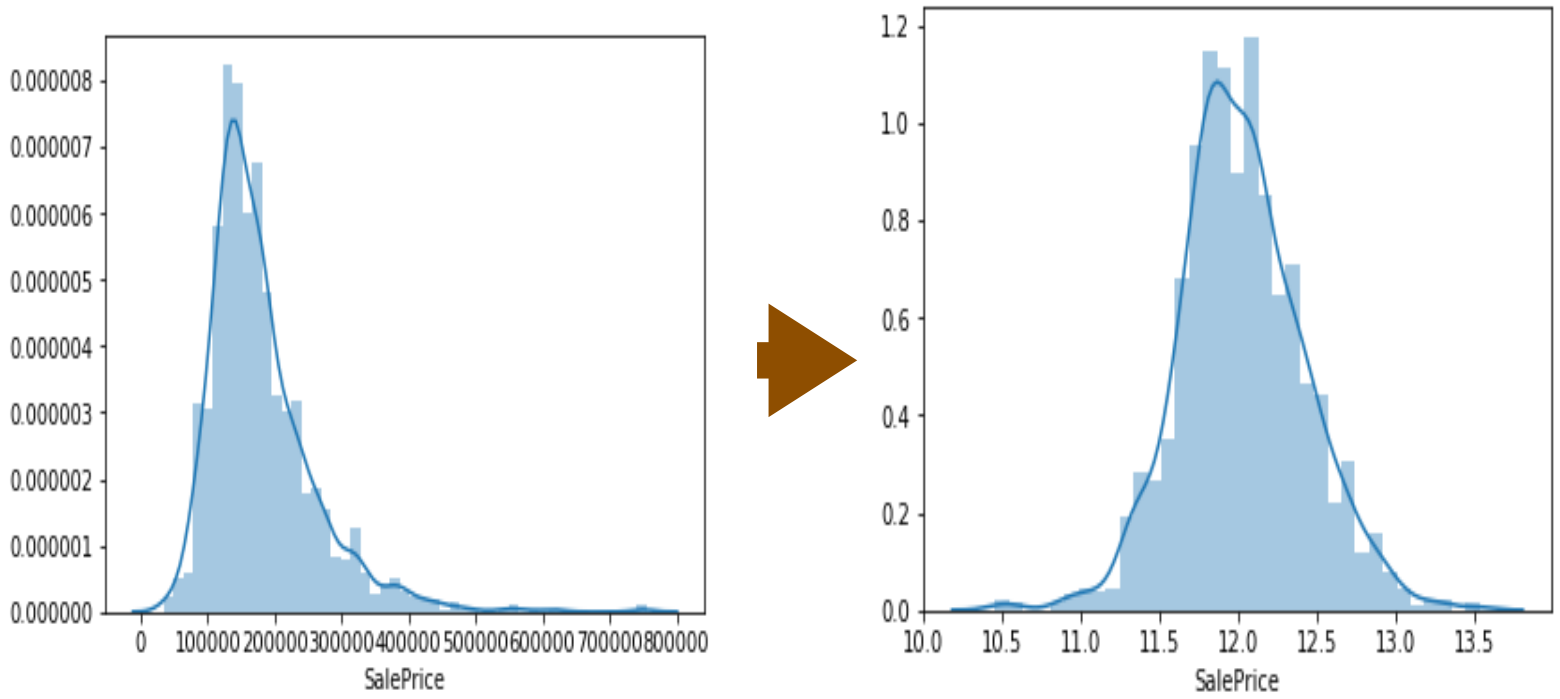
Standardizing Numeric Data

by taking the mean of each feature, subtracting it from each entry and dividing it by standard deviation.



Dataset Preprocessing: Handling SkewedData

- The Sale price was skewed right and to make it symmetric we took log transform.



Dataset Preprocessing: Categorical Data to Dummies

- Converting categorical data into numerical form.
- Replacing missing values by baseline dummy values.
- We handled the features of conditions and exteriors by converting their categorical data to numeric form by placing dummies.
- Data type used for this purpose was float.

Dataset Preprocessing: Split Train & Test Validation Sets

- Split the Dataset randomly into Train and Test sets.
- Train/Test split was 90% & 10% respectively.

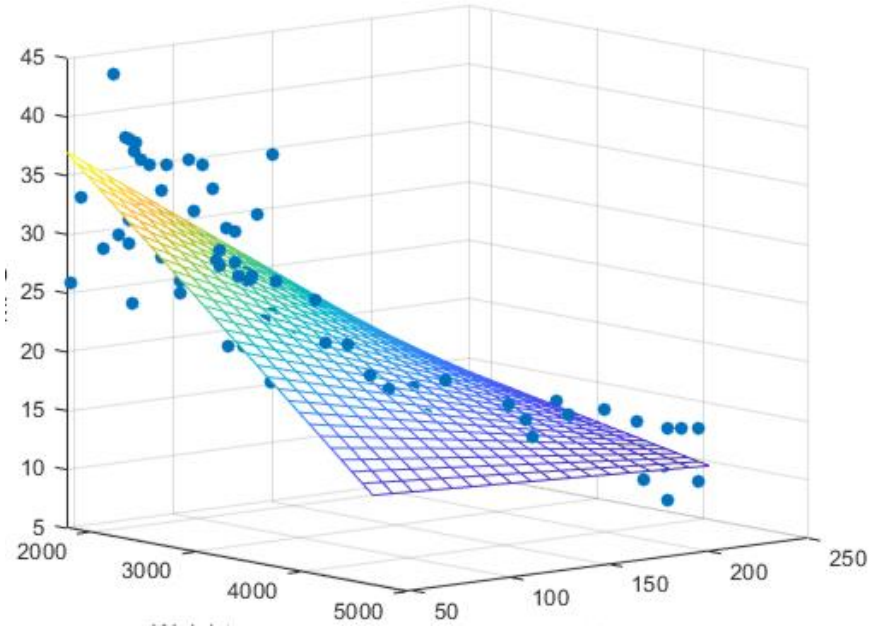
ALGORITHMS

Regression Algorithms

1) Multivariate Regression
2) Gradient Descent
3) Stochastic Gradient Descent
4) Ensembled Learning
5) Random Forest Regression
6) Multi-Layer Perceptron
7) Kernel Ridge Regression

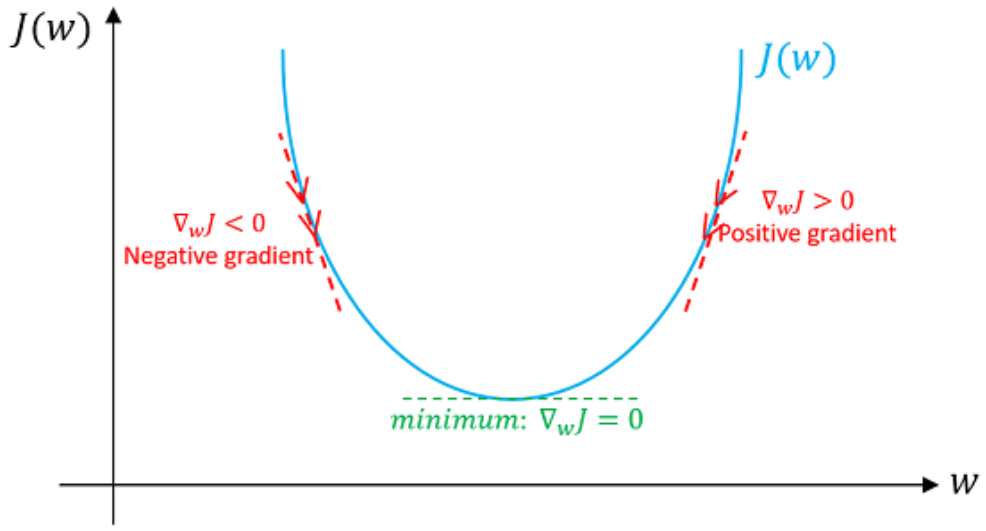
1) Multivariate Regression

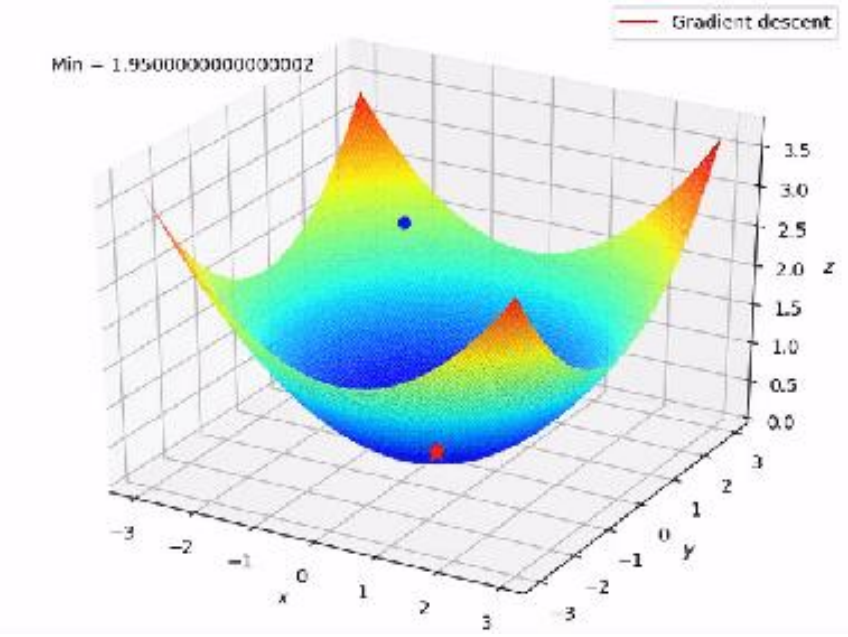
- Learns linear decision boundary using normal equation.
- Provides optimal solution for linear modeling.
- Best in cases where data set is not very large as computational complexity is reduced.
- For very large datasets, space complexity cripples the model.



2) Gradient Descent

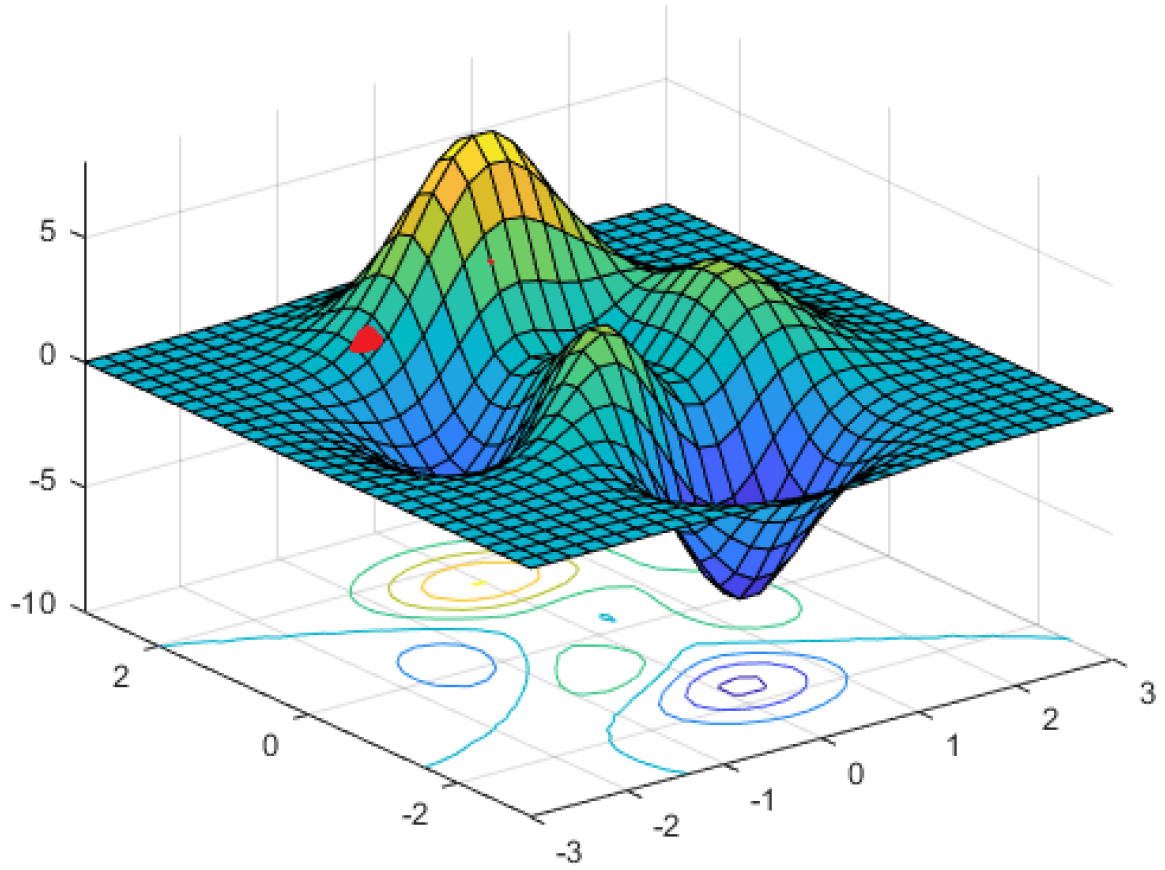
- It is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function (cost).



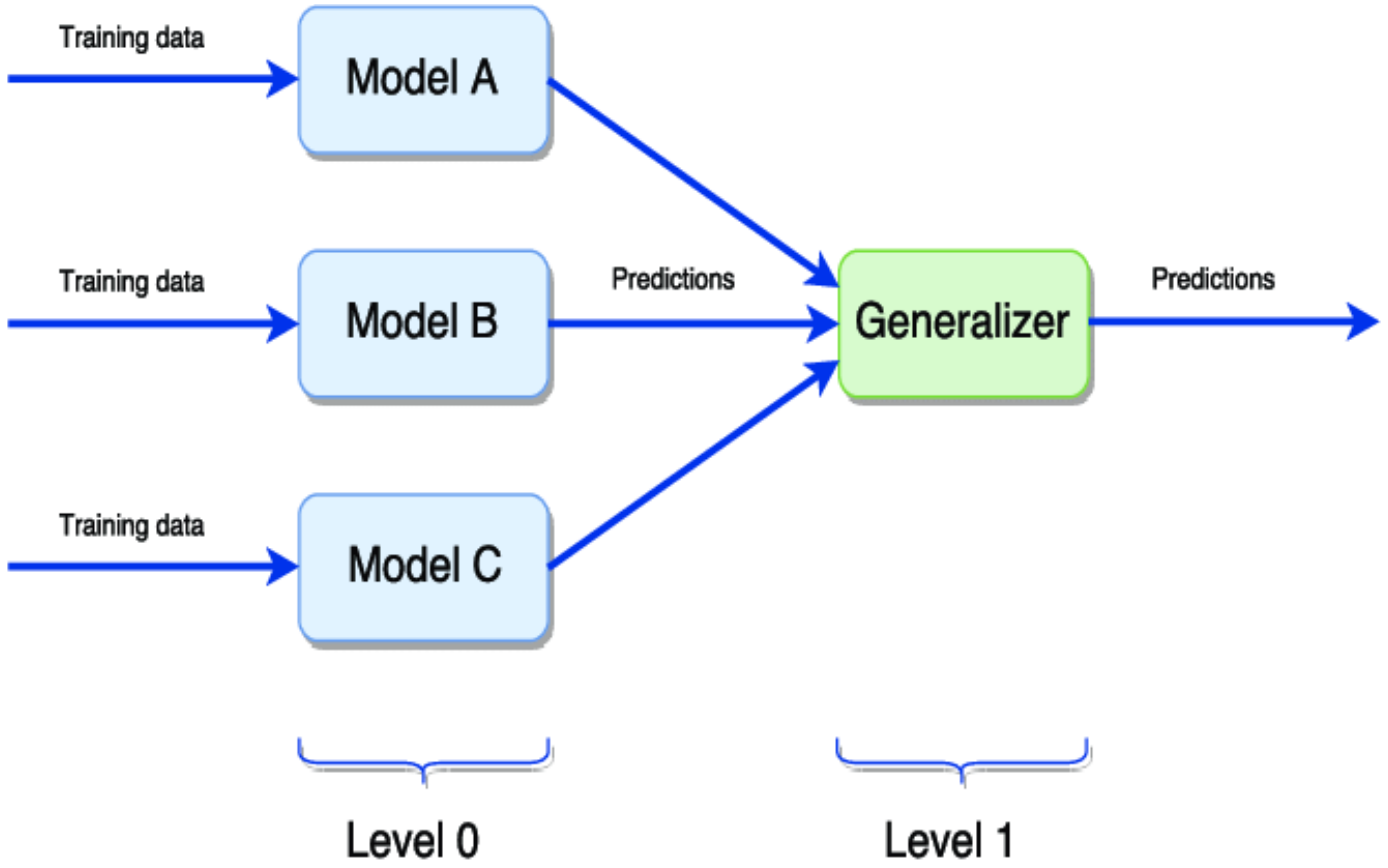


3) Stochastic Gradient Regression

- Stochastic gradient descent performs better than batch gradient descent.
- Better at finding global minimums as compared to local minimums.

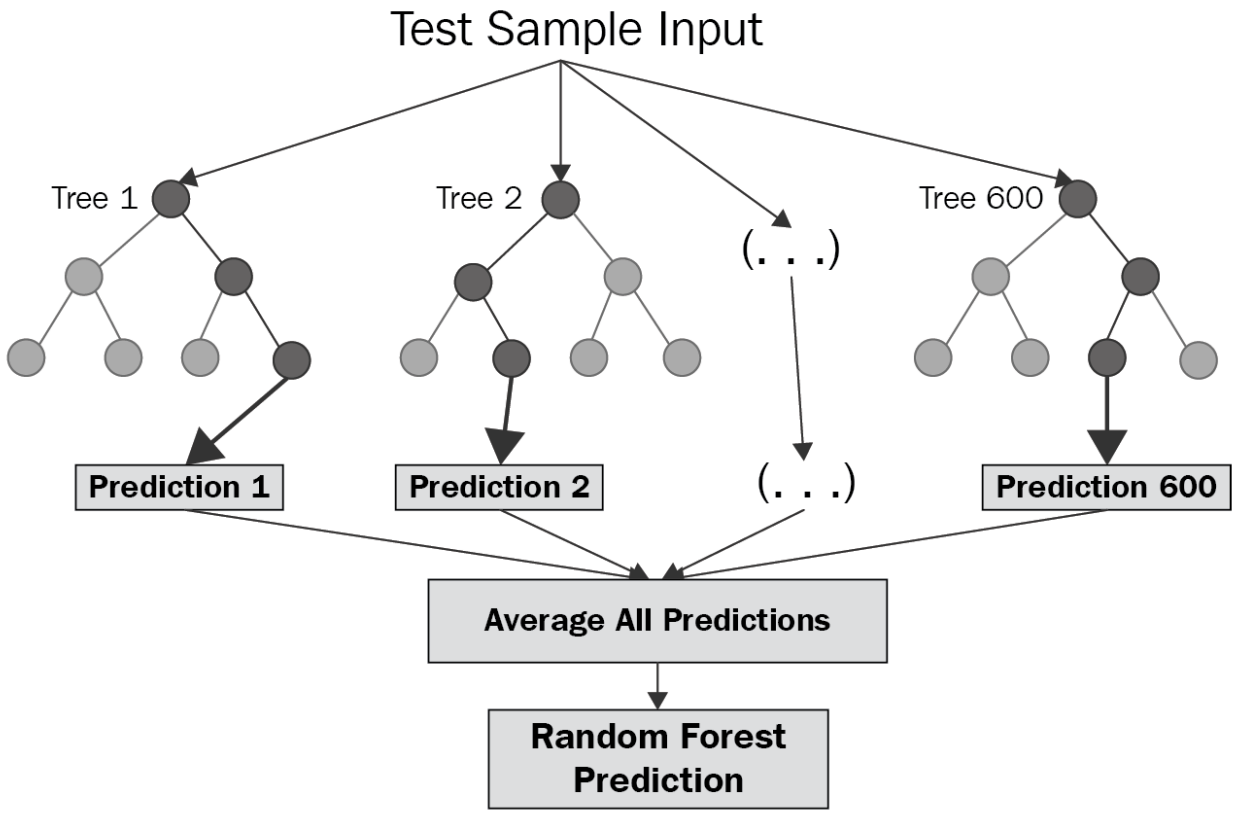


4) Ensembled Leaning



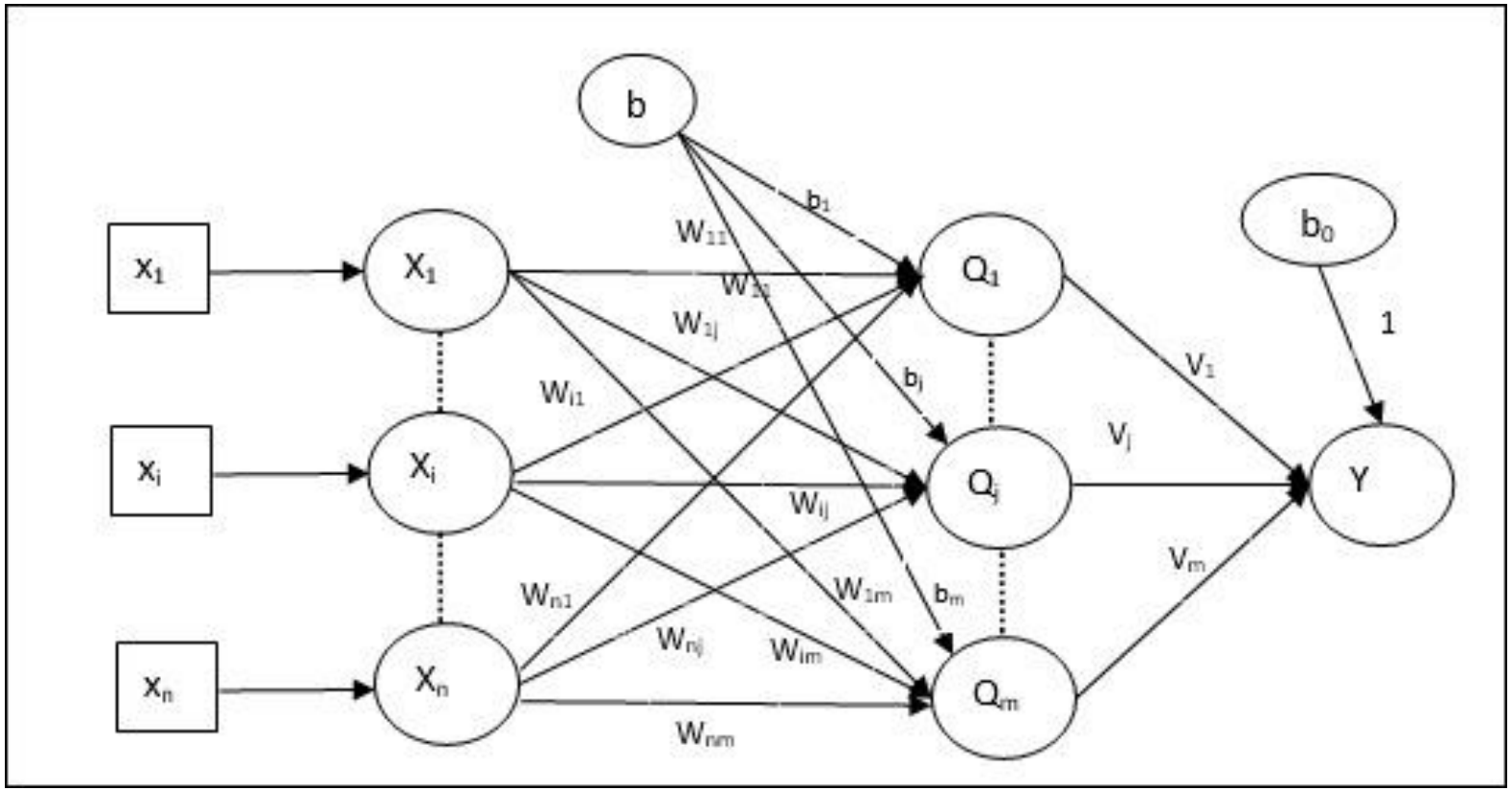
5) Random Forest Regression

- Enhances decision tree learning.
- Learns multiple decision trees on different feature splits.
- Ensembles the results for regression.
- Avoids overfitting by not relying too heavily on specific features.



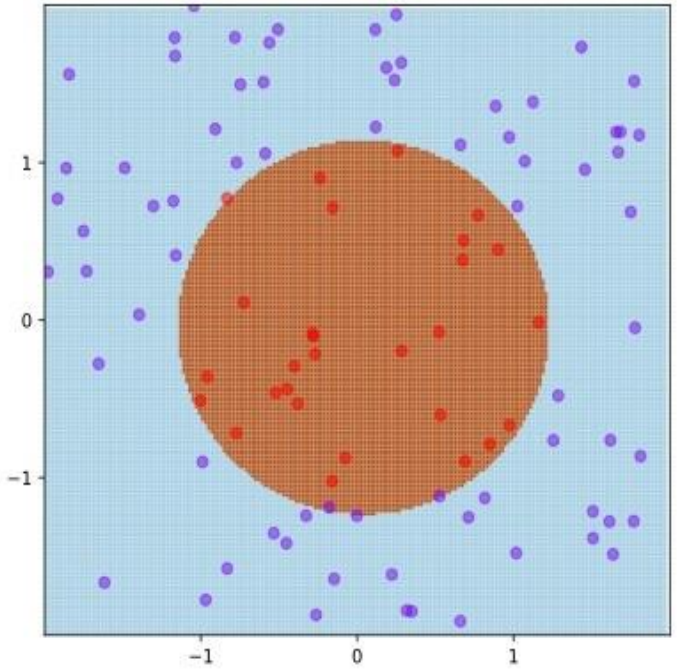
6) Multi-Layer Perceptron

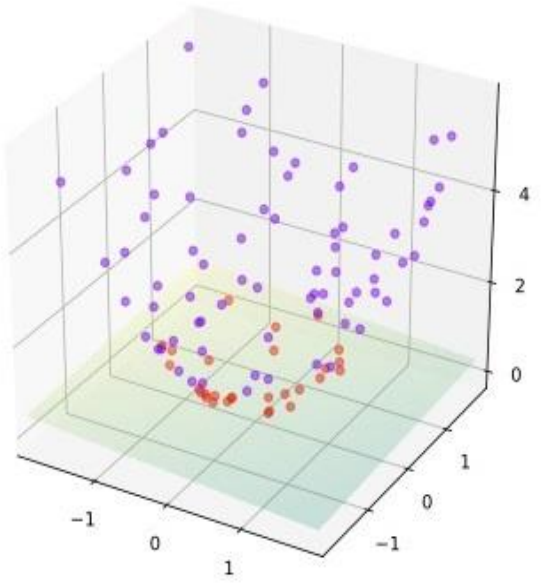
- Multiple hidden layers.
- Linear activation.
- Backpropogation.
- Learns non linear decision boundary.



7) Kernel Ridge Regression

- Combines kernel trick with Ridge regression.
- Maps data points to higher order domain inexpensively.
- Linear hyperplane in higher order domain is indeed a nonlinear decision boundary in lower order domain.

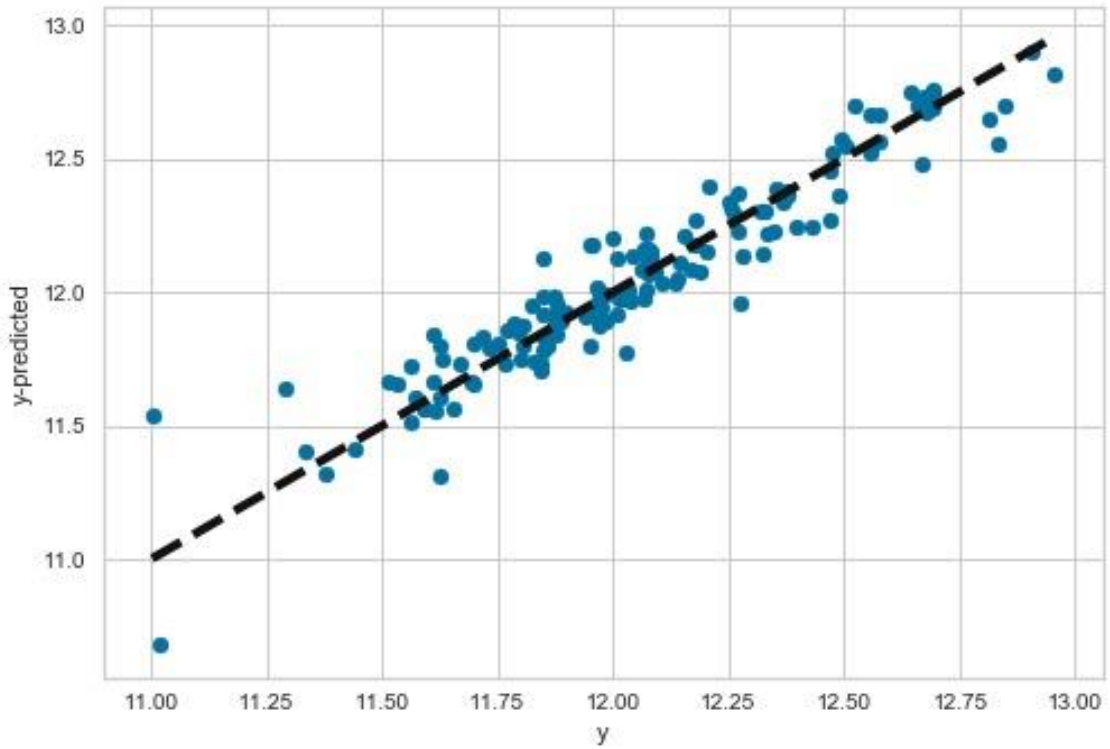




RESULTS & ANALYSIS

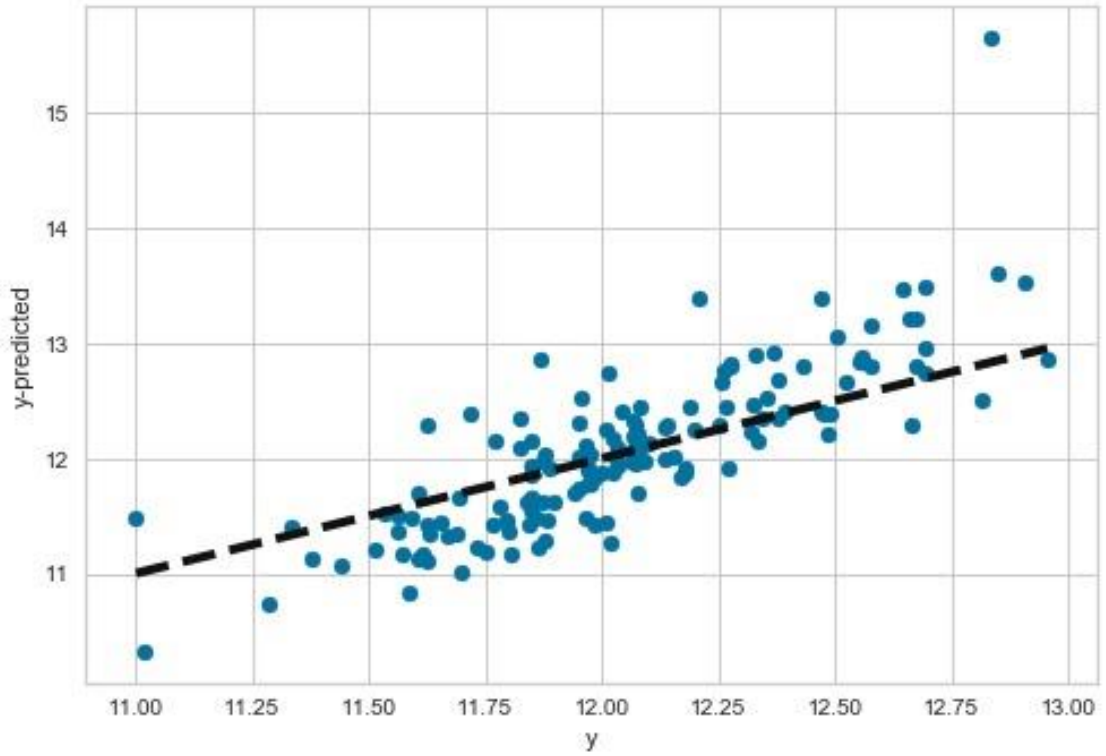
RESULTS: Multivariate Regression

- The results are reasonable.
- RMSE: 0.1206
- Better as compared to Batch & Stochastic Gradient Descent.
- Optimal Linear Decision Boundary learned but can't be compared to non-linear model.



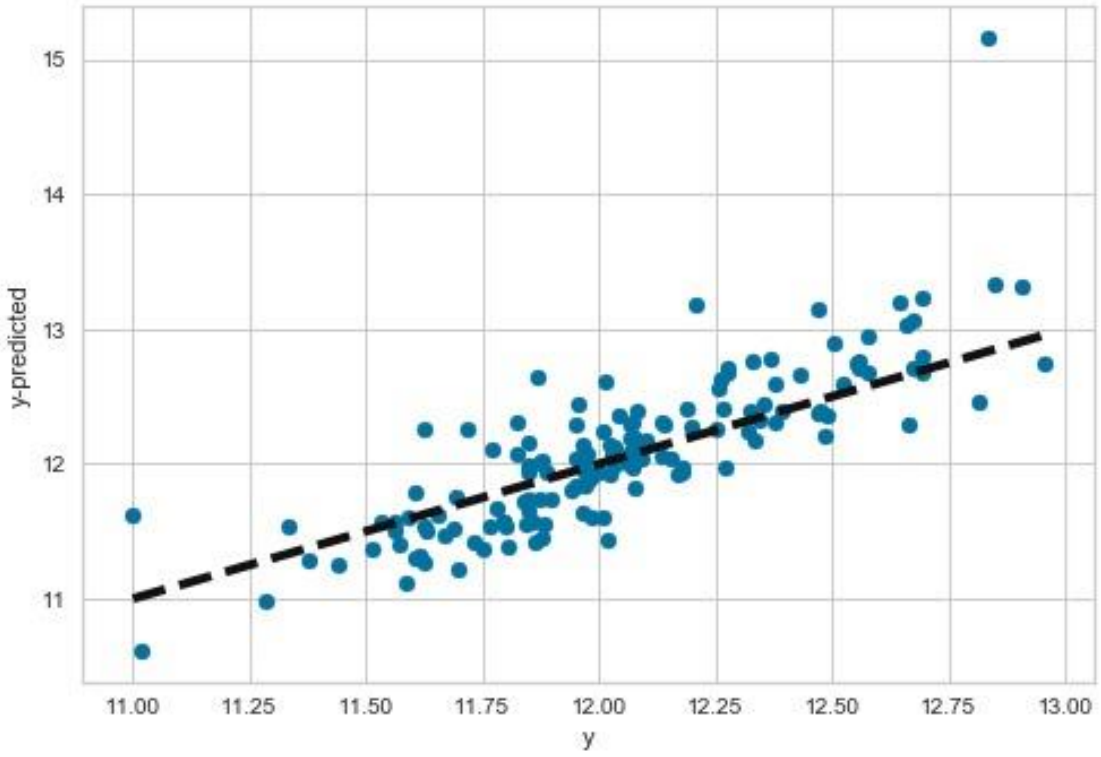
RESULTS: Gradient Descent

- The model performed the worst as compared to others.
- RMSE: 0.4502
- Batch processes are more prone to settling at local minima rather than global minima.



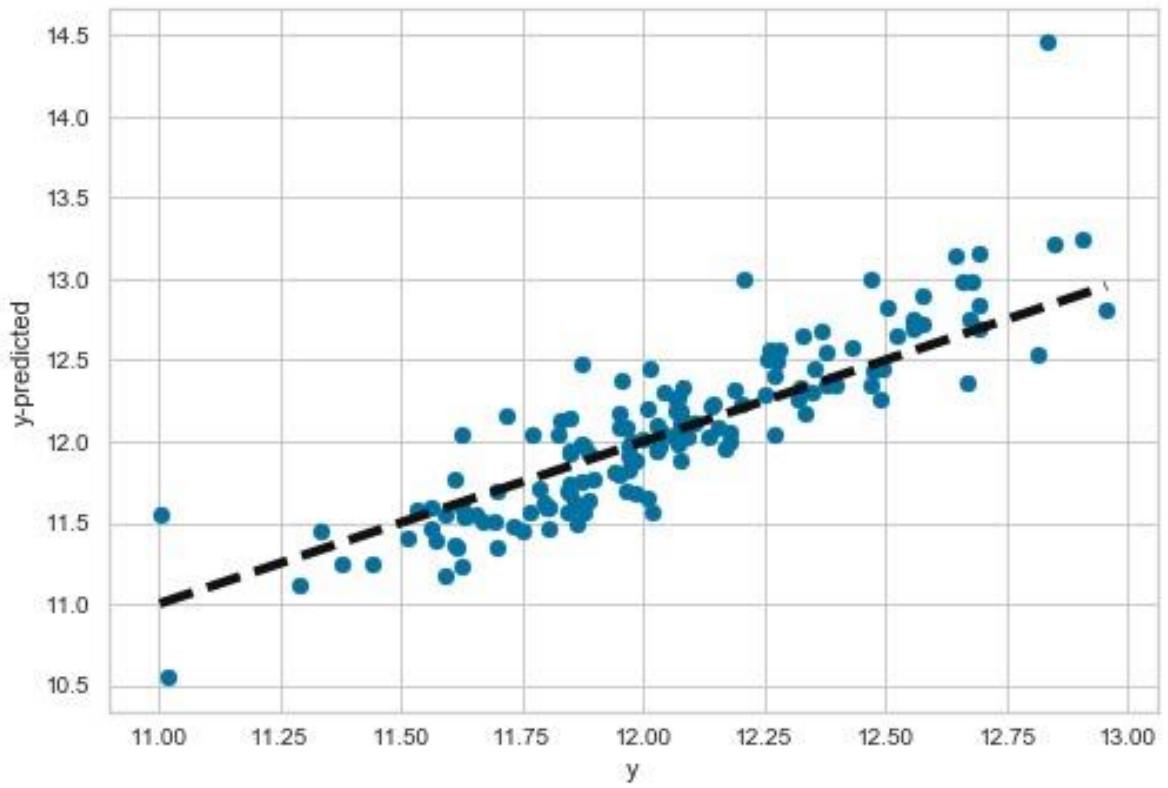
RESULTS: Stochastic Gradient Descent

- The model performed better than Batch Gradient Descent.
- RMSE: 0.3451
- Stochastic processes iteratively modify weights giving them a high chance of finding global minima.



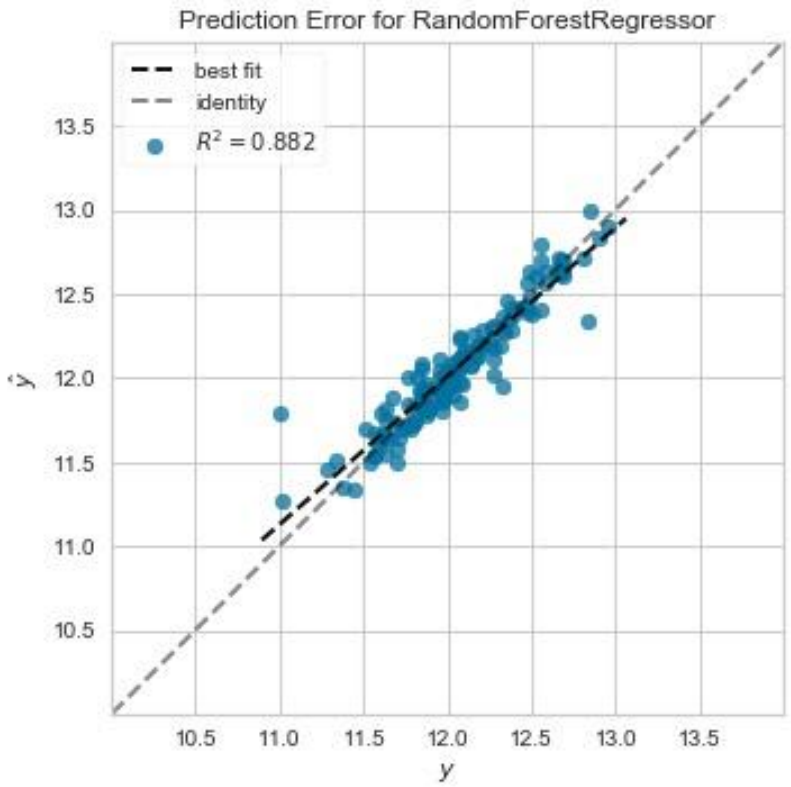
RESULTS: Ensembled Learning

- The results were in between of 3 of our linear models.
- RMSE: 0.2695
- Ensembling shows that different models can be combined to produce better results.



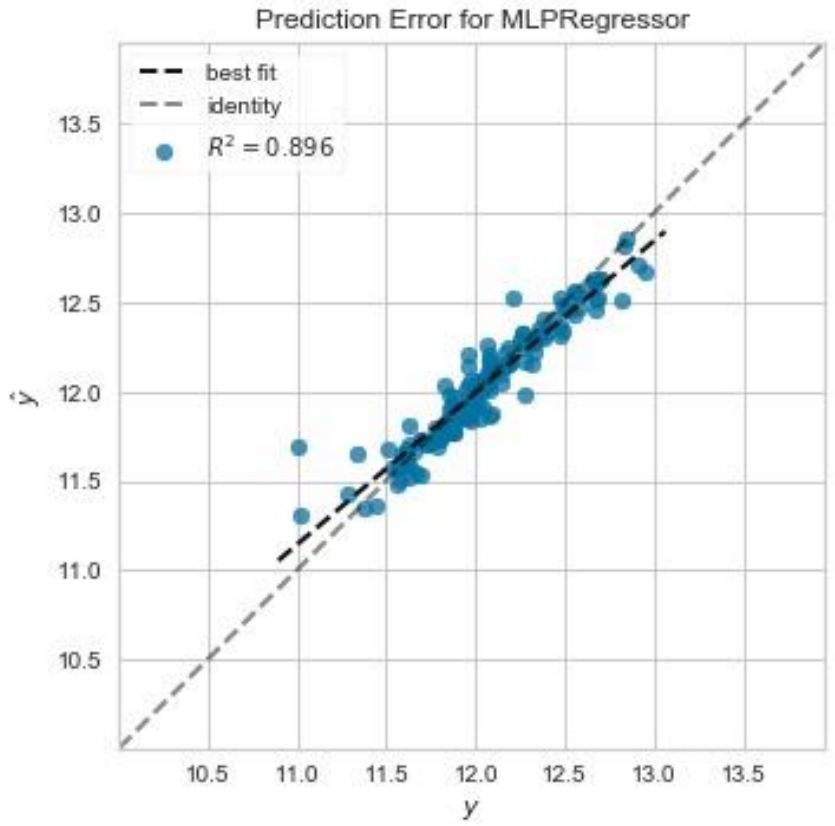
RESULTS: Random Forest Regression

- There is a sudden jump in accuracy as we shift to non-linear models.
- RMSE: 0.1256
- Random Forest combines multiple decision trees to produce an accurate averaged non-linearly regressed output.



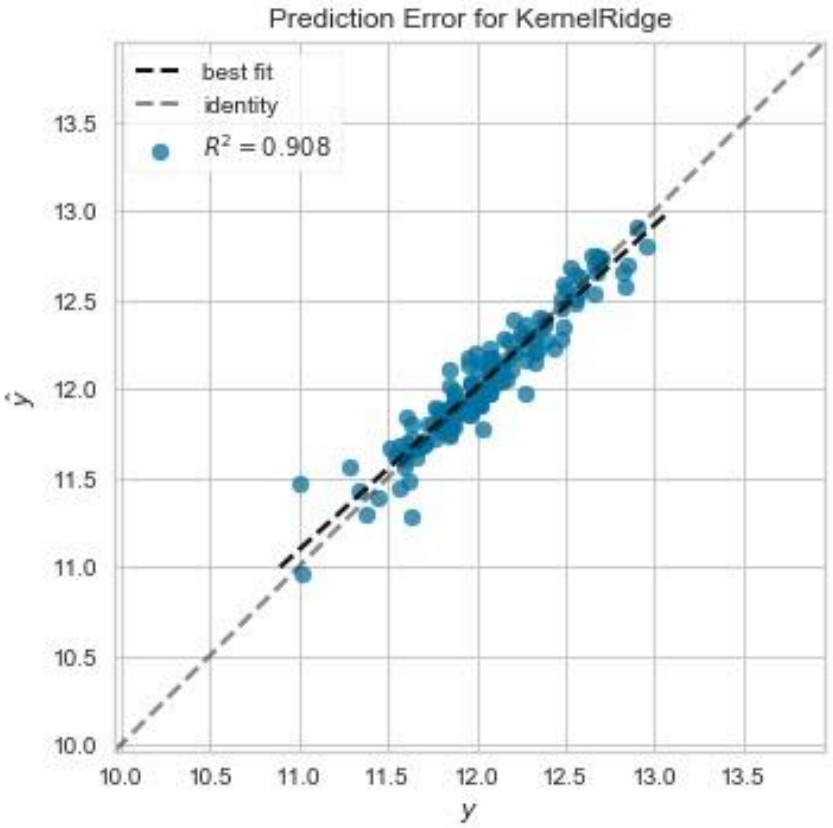
RESULTS: Multi-Layer Perceptron

- A combination of multi-layered perceptrons including hidden layers enable the model to learn a non-linear function.
- RMSE: 0.1195
- Better than all of the models except Kernel Ridge Regression.



RESULTS: Kernel Ridge Regression

- The model greately enhances the optimal linear multi-variate regression by mapping the data points to a higher order domain and learning a linear function which eventually results in a non-linear function in a lower order domain without any increased computation expense.
- RMSE: 0.1126
- OUTPERFORMED



CONCLUSION

- Non linear methods like ANN, Kernel Ridge and Random Forest perform better than linear models like Multivariate and Gradient Descent Regression.
- The results back our conclusion as the RMSE fell to 0.1126 in Kernel Ridge while Batch Gradient Descent had 0.45 which is a big difference.
- We can conclude that for standardized variables with reasonable sized datasets, Kernel Ridge performs the best.
- In cases, where datasets are very large the Kernel Perceptron will inevitably be the best possible solution as the space complexity of Ridge Regression will be very high.


Thank
you!!