Improving Accuracy of Real Estate Valuation Using Stacked Regression

[1]Dhvani Kansara, [2]Rashika Singh, [3]Deep Sanghvi, [4]Pratik Kanani

[1-3] Students, [4]Assistant Professor
[1]Information Technology Engineering,
[1]Dwarkadas. J. Sanghvi College of Engineering, Mumbai, India

Abstract— **Real Estate business is flourishing with each passing day making it imperative for an effective house prediction model. This in turn will be beneficial to both the investors as well as the estate owners who can get effective prices without depending on external third party agents or mere capitalization rates. Machine Learning can be leveraged for this purpose. This paper evaluates the algorithms which can be used to predict the house prices on the Boston dataset [1] taken from Kaggle with 79 attributes like Living area, Condition at time of sale, Proximity to roads and rails, year built, etc. taking into consideration all aspects from homes in Ames, Iowa. In order to predict House Price, which is the dependent variable, regression algorithms under supervised learning are used. This paper provides the most optimal solution with the Stacked Regressor, an ensemble model which in this case averages Multiple Linear regression, Random Forest Regression and XGBoost regression algorithms ultimately giving a Root Mean Square value of 0.1047 and a high accuracy of 93.52.**

Index Terms— **Real Estate Price Prediction, Regression algorithms, Model Stacking**

I. Introduction

Property valuation is an important task for parties in various spectrums of real estate business involving developers, lenders, brokerage owners as well as investors. It is necessary for them to gauge the risks in investments and assure whether the projected returns will match their expectations from investing in housing developments. These parties mostly rely on external valuation or internal modelling methods that use excel or other mathematical formulae based on comparable recent sales driven by capitalization (cap) rates [21]. These measures are unreliable and imperfect as they do not imbibe the property's all unique characteristics into consideration during calculations due to ample number of parameters involved. Evaluating these prices efficiently is important to achieve a fair negotiation and to determine lending limits and transaction prices. Due to the growing investments in real estates much advanced prediction is required. This is where machine learning comes into picture. Machine learning is a branch of artificial intelligence which aims at automatically learning patterns of data from experiences without being explicitly programmed to develop a predictive model. Thus, Machine Learning (ML) models determine value by comparing attributes of properties transacted in the past, and market conditions at the time of those transactions, to the attributes and timing of the target, and are evidently not based on cash flow models [16]. Thus these models possess the ability to consider the entire range of parameters to analyse patterns to give an accurate result in an easier and faster manner. The prediction models continuous-valued functions i.e., predicts new values. Also ML models can be used and applied on traditional computing techniques to improve the existing accuracies [11-15].

Supervised learning [2] uses data with existing class labels which provide output based on the input pairs whereas in unsupervised learning the class labels are not known. Our target is to predict real estate pricing, which are continuous valued numbers, thus regression is used to solve this problem. Regression is a supervised learning technique which uses a model to predict ordered values from existing data. It calculates dependent variable value based on one or more independent variables [3,4]. Regression can be broadly classified into three types-linear and multiple regression, non-linear regression and other regression. The dataset being used is taken from Kaggle called the Boston dataset with 79 variables including building type, utilities, size, rooms among others which might contribute to the pricing of the houses. This data is pre-processed and trained using three regression algorithms namely Multiple Linear Regression, Random Forest Regressor, XGBoost and Stacked Regressor - formed by combining above three algorithms, which is an ensemble model [4] used for obtaining better accuracy. These models can be used in real estate business as well as by consumers for checking the real estate price if they plan on investing without being dependent on external agents.

In this paper, the steps involved, right from cleaning, pre-processing, feature engineering, training and modelling the dataset to obtain a high accuracy score of 93.52 from the stacked regression model are presented.

II. Methodology

2.1 Pre-Processing

The dataset consists of 79 independent variables each contributing to the final estimate by a certain factor. This step is essential in order to analyse the available data and make it appropriate for applying machine learning algorithms on them, so as to obtain the best possible accuracies during modelling. In order to do this, data cleaning and feature engineering is employed.

2.1.1 Cleaning

By analysing the data using scatter graphs of each variable with respect to the target attribute, it was checked if the dataset has any outliers, if so, they were removed, as shown in fig. 1(a) and 1(b).

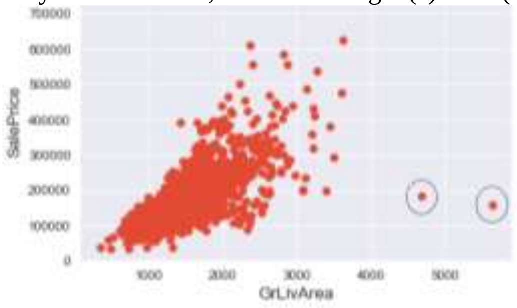


Fig. 1(a): Outliers in dataset

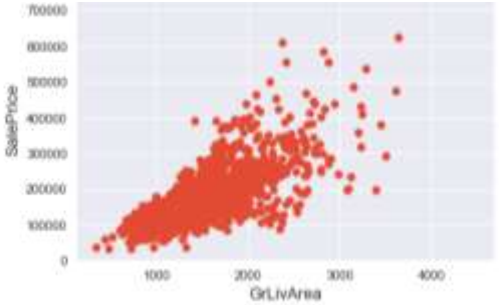


Fig. 1(b): Outliers removed

An outlier is a data point which lies far away from other observations, these observations can skew or mislead the final predictions ultimately reducing the accuracy of the results. Next, after analysing the target variable-Sale Price, it was observed that their distribution is skewed towards the right as seen in fig. 2(a). Most machine learning algorithms are designed to perform on normalized data, reason being- Homoscedasticity [5] i.e. in order to ensure that our model does not commit small errors for low values and big errors for higher values of target feature, this is performed. In efforts to make difference between predicted and true values constant and make sure errors the model commit have the same variance, this task is performed. This is ensured by making the target attribute follow a gaussian distribution. Thus log transformations are applied in order to normalize the sale price. The normalized results are as shown in fig. 2(b).

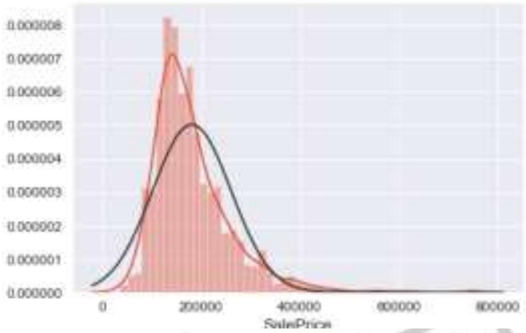


Fig 2(a): Right skewed data

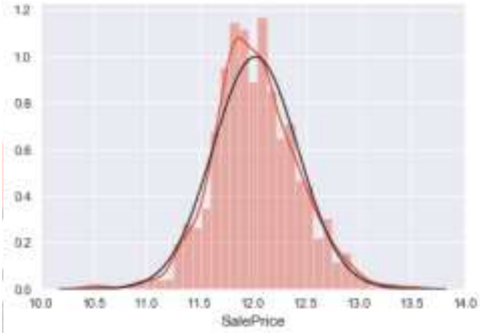


Fig 2(b): Normalized data

2.1.2 Feature Engineering

After proceeding sequentially through the data, there occurs imputing missing values for each variable and filling them with appropriate values. These values were filled based on the contexts of the attributes. The numeric missing values like ‘area of lot frontage’ were replaced by median of the neighborhood lot frontages. The categorical missing values were filled by adding a new category called ‘none.’ Certain other categorical missing values like type of electric supply, kitchen quality, type of sale, were replaced by the most recurring value of that column i.e. by the mode. Scaling is performed on the dataset, i.e. to make all the features represent by a standard scale. This is because, if the measures of different features are done on wildly different scales then this will reduce the model’s ability learn because there is no standardization, some features may not be relevant but add more weight due to higher scale and others with small values may not contribute significantly even if they are important. Scaling will ensure standard feature values weighs all features equally in their representation.

Next, it was observed that certain variables are categorical but have been assigned numeric values. These were transformed. These categorical values are converted into binary data which will make sense to the model. It is required by the models to interpret these attributes’ contribution to the final prediction. For this, label encoding and one-hot encoding is applied. One-hot encoding creates a new column for every categorical value and assigns 1 if a particular row has that value or 0 otherwise. After this step, there are 220 attributes in our dataset, all cleaned and engineered.

2.2 Algorithms Used

2.2.1 Multiple linear regression

Linear regression is a common way of predictive analysis where a target variable is predicted based upon certain independent input variable. Unlike Linear regression, Multiple Linear Regression [6] models uses linear relations between multiple predictor variables. It works by assigning certain weight to each of the parameters which contribute to the final outcome.

A multiple linear regression model with k predictor variables X1, X2, ..., Xk and a response y, can be written as in Eq. 1.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_k X_k + \varepsilon \qquad (1)$$

where ε is the residual or the error parameter and β is the weights assigned to each parameter.

Multiple linear regression also takes into consideration the interaction effects of two or more predictor variables. Each coefficient is interpreted as estimated change in y corresponding to a one unit change in a available, when all other variables are constant.

2.2.2 Random forest regressor

A decision tree is a model that uses tree like graphs to predict possible outcomes. While constructing the graph, it takes into consideration parameters like cost, constraints, benefits etc. to form a rule based tree whose leaf nodes contains the result. Random forest [7] is a type of an ensemble, which means a collective prediction of several algorithms. This algorithm is an adaptation of decision trees, where a model makes predictions based on a sequence of base models which can be expressed as in eq. 2.

$$g(x) = f_0(x) + f_1(x) + f_2(x) + \ldots f_k(x) \qquad (2)$$

f_k represents k decision tree regressors, and where each base model is a decision tree.

All base models are constructed independently using a different subsample of the data. Each tree is built using a logic such that, a splitting attribute is determined from where the tree will branch, the selection of this attribute is done on the basis of calculation of standard deviation. Standard deviation from the actual result is calculated based on one split point and the attribute which will result in minimum standard deviation is chosen as the split point. Prediction of continuous value is made based on the average of output values of all regression trees in a forest.

Random forest is very good at handling tabular data with numerical features [7]. Unlike linear models, such as linear regressors, it can take into consideration the nonlinear interaction between the features and the target.

2.2.3 XGBOOST

XGBoost stands for extreme gradient boosting. It is an implementation of gradient boosted decision trees. This model works by adding models on top of each other iteratively where the errors of the previous model are corrected by the next predictor, until the training data is accurately predicted or reproduced by the model. Weights are assigned to each model during the intermediate steps based on Root Mean Square (RMS) calculations with the aim to minimize the RMS. Thus, it will pay more attention to previously wrongly predicted data by assigning larger weight to it so that such data can be learnt effectively. A learning rate is defined which states how hard each tree tries to correct the error of the trees above it. The model is assigned a learning rate of 5%, i.e. models will be assigned weights in order of 0.05. The number of boosting stages (estimators) i.e. number of times the algorithm runs with different samples is chosen to be 2000. The maximum depth of each regressor estimator is set as 3. XGBoost is suitable is used for improving the execution speed and the model performance [7].

2.2.4 Stacked regressor

Stacked regressor is an ensemble model [9] formed by a linear combination of multiple prediction algorithms to improve the overall prediction accuracy. The optimal coefficients in this linear combination are determined by using least squares under non-negative constraints and cross-validation.

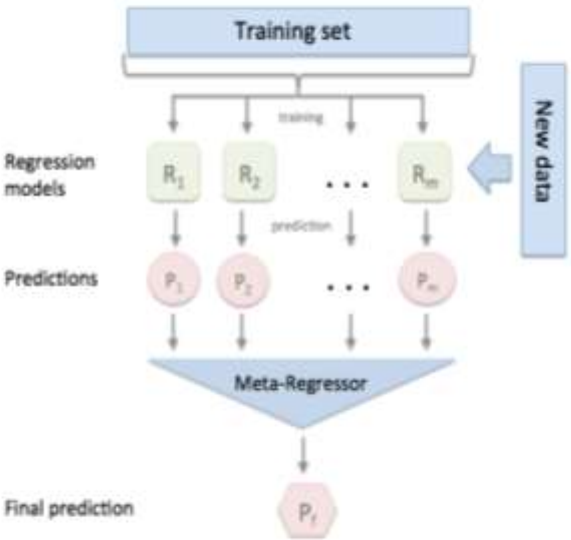


Fig.3: Stacked regressor [8]

Our model uses stacking of multiple linear regressor, random forest regressor and XGBoost regressor. The Stacked Regressor works by selecting first level regression algorithms which are fit into the training data which will give a set of outputs. These output predictions are fed as inputs for the second level regressors and are fit using the data training data. The model is trained using out-of-fold sampling. In this, the dataset is divided into k folds, where k-1 folds are fit into first level regressor and then to check its performance, it is tested on remaining one left fold. This occurs for all data in the train set. These results are then provided as input to second level regressor. This gives a new set of predictions as output. The optimization is done based on minimizing least square errors.

Stacked regressor is used in order to build a strong model which takes predictions of other diverse and well-chosen algorithms into consideration to provide the final output. Each algorithm makes a significant contribution and bias/weakness of any model is offset by strength of other models, thereby elevating the overall accuracy [16]. Because of rigorous cross validation the inaccuracies resulting from one model can be handled by output from the other models thus minimizing error rates by balancing.

III. EXPERIMENT

The dataset is obtained from kaggle website [1]. It consists of 80 attributes, 79 of which represent all the factors which may contribute to the price estimate of a house and the last attribute is the price of the corresponding house. The aim is predicting real estate values pertaining to the current market value based on all these factors. Regressions algorithms are applied to calculate this price. In order to learn and evaluate the model, the dataset was divided into two parts. One part is called training set which constitutes of 80% of the data points and remaining 20% forms the test set. The training set is used to train the model, that is, using the values in this set the model will learn changes in price estimate (dependent variable) with respect to changes in independent variables. The test set is used to evaluate how correctly the trained model is predicting on new values. Thus the sale price in the test set were compared with the sale price predicted by our model, the difference between these two values are calculated to evaluate the regression performance.

IV. IMPLEMENTATION

In order to perform the experiments, platform used was spyder as IDE to code in python 3.0. For making predictions with least errors, pre-defined scikit-learn libraries in python were used which are built using numpy, scipy and matplotlib [10]. For implementing the Stacked Regressor mlxtend library [20] defined in python was used. Since our aim is to perform regression, which is based on continuous data as output, root mean square error (explained in section **V**) is used as a measure of fit. Fig. 4 shows the implementation of our model stacking 3 different regression algorithms. Where Multiple Linear Regressor and Random Forest Regressor form the first level regressors whose output are fed into XGBoost regressor model to obtain new predictions.

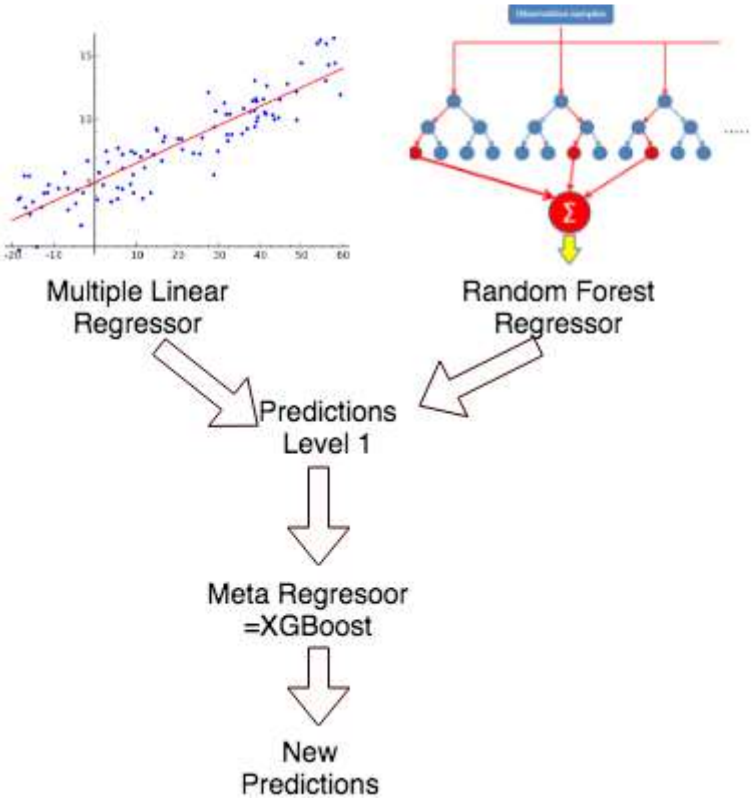
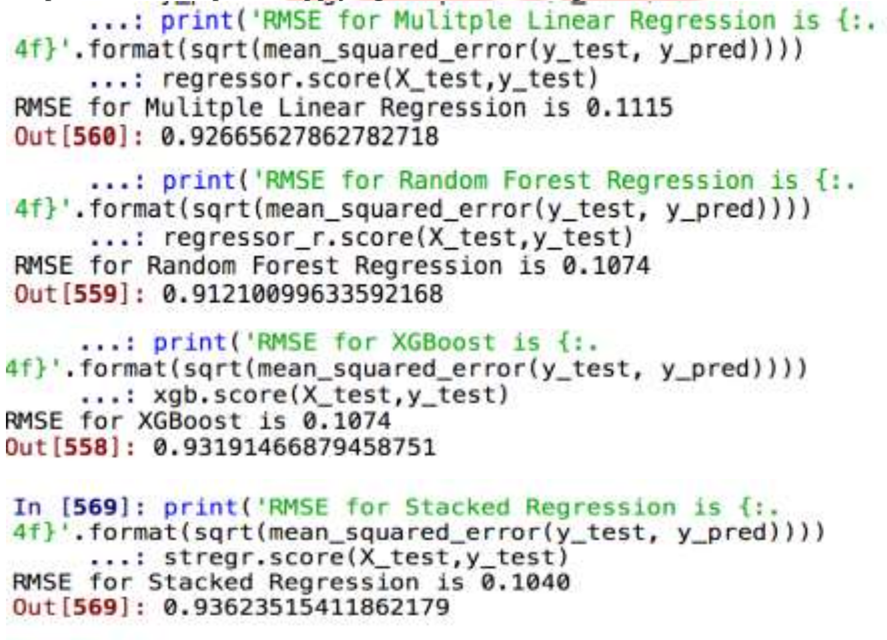


Fig. 4: Stacked Regressor Model Design

The figure 4 below represents the outputs of applying the mentioned algorithms on the dataset.



```
    ...: print('RMSE for Mulitple Linear Regression is {:.
4f}'.format(sqrt(mean_squared_error(y_test, y_pred))))
    ...: regressor.score(X_test,y_test)
RMSE for Mulitple Linear Regression is 0.1115
Out[560]: 0.92665627862782718

    ...: print('RMSE for Random Forest Regression is {:.
4f}'.format(sqrt(mean_squared_error(y_test, y_pred))))
    ...: regressor_r.score(X_test,y_test)
RMSE for Random Forest Regression is 0.1074
Out[559]: 0.91210099633592168

    ...: print('RMSE for XGBoost is {:.
4f}'.format(sqrt(mean_squared_error(y_test, y_pred))))
    ...: xgb.score(X_test,y_test)
RMSE for XGBoost is 0.1074
Out[558]: 0.93191466879458751

In [569]: print('RMSE for Stacked Regression is {:.
4f}'.format(sqrt(mean_squared_error(y_test, y_pred))))
    ...: stregr.score(X_test,y_test)
RMSE for Stacked Regression is 0.1040
Out[569]: 0.93623515411862179
```

Fig. 4: Results of applying each algorithm on the dataset

V. Results

The prediction takes into consideration three algorithms to find out the model with the best accuracy in order to create a system for faster prediction. The regression model is evaluated based on the following metrics: metrics Root Mean Squared Error (RMSE) and R Squared (accuracy)[17].

Root Mean Squared Error [18] is calculated as the square root of Mean Square Error which is the difference between the values predicted by a model and that actually observed. RMSE is how spread out the predictive errors are. The RMSE value is inversely proportional to the performance, the lesser value the better model with zero indicating a perfect fit, since the predicted and actual value have a difference of 0 indicating they are same.

This formula as given in Eq. 3 evaluates the root mean square error between the logarithm of predicted value and observed value of salesprice.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (3)$$

Where y_i = log(true/actual value of salesprice), $\hat{y}_i$= log(predicted value of salesprice) and n = total number of observations

Another metric used defines the accuracy with respect to regression line produced. This is called R-squared. It measures the deviation of all the results from the fitted regression line [19]. It is directly proportional to performance, the higher values the better is the performance of the model as the higher the values smaller is the difference between observed and fitted values. The formula is given in Eq. 4.

$$R^2 = \frac{Variance\ explained\ by\ the\ model}{Total\ Variance} \qquad (4)$$

Thus, a 100% variance indicates that the model explains all the variation around the mean regression line.

Table 1: Results of trained models on testing data

Algorithm	Root Mean Square Error	Score Accuracy (R-squared)
Multiple Linear Regression	0.1115	92.66
Random Forest	0.1217	91.19
XGBoost	0.1076	93.17
Stacked Regressor	**0.1047**	**93.52**

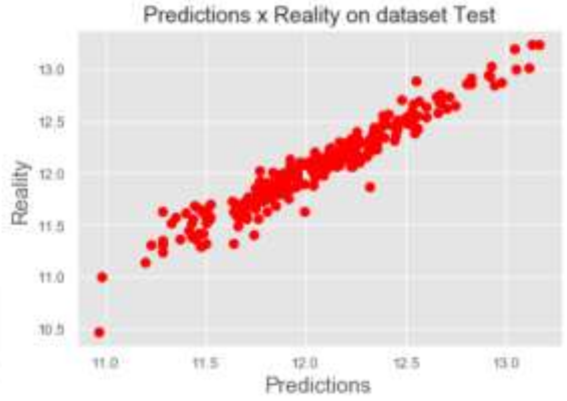


Fig.5: Graph representing actual prices vs. prices predicted by our model on test data using stacked regressor

The graph in Fig.5 represents an almost linear relationship between the actual and predicted prices showing a very well fit around the regression line. Based on our observations, the Stacked Regressor provides a better optimal prediction with a RMSE value of 0.1047 and a score accuracy of 93.52.

VI. Conclusion

This study performs various empirical tests on the Boston dataset from Kaggle with 79 variables for house price prediction in real estate business. The results from the table in above section show that the Stacked Regressor provides the most optimal solution with the Root Mean Square Error value of 0.1047 and Score Accuracy of 93.52. Thus, saying that multiple trained models learned on different regression algorithms can be cross validated averaged to pass into a meta regressor to elevate the accuracies of individual models so as to effectively predict real estate values. Predicting house prices more accurately will benefit not only the developers but also other investors and prospective homeowners to encourage them to make investments and associate the risk factor. Using pre-processing, training and feature engineering the most optimal solution for the problem at hand was found using the Stacked Regressor, which can be leveraged for forecasting house prices in real estate business effectively.

VII. Future scope

The current size of the dataset used contains 1500 rows, this size can be increased to further improve the accuracy using the stacked regressor. This involves collecting more data which will enable training more exhaustively. If there is more data, one can also apply deep learning algorithms which will abstain us from the task of feature engineering as the model itself will learn to choose attributes efficiently. Principal component analysis can also be performed on our cleaned data such that out of 79 the most important features which actually contribute to the final output can be selected to perform analysis on.

REFERENCES

[1] Kaggle housing prices (2018,August,12). Kaggle Inc. [Online] Available: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

[2] Machine learning algorithms in human language (2018,August,13) Datakeen [Online] Available: https://www.datakeen.co/en/8-machine-learning-algorithms-explained-in-human-language/

[3] Regression (2018,August, 13) Business Dictionary [Online] Available: http://www.businessdictionary.com/definition/regression.html

[4] Stacking Regressor (2018,August,14) Github [Online] Available: https://rasbt.github.io/mlxtend/user_guide/regressor/StackingRegressor/

[5] Homoscedasticity(2018,August,15)Wikipedia [Online] Available: https://en.wikipedia.org/wiki/Homoscedasticity

[6] Multiple Linear Regression(2018,August,15) Cornell.edu [Online] Available: http://mezeylab.cb.bscb.cornell.edu/labmembers/documents/supplement%205%20-%20multiple%20regression.pdf

[7] Random Forest Regression(2018,August,15) turi.com [Online] Available: https://turi.com/learn/userguide/supervised-learning/random_forest_regression.html

[8] A Gentle Introduction to XGBoost for Applied Machine Learning(2018,August,16) machinelearningmastery [Online] Available: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/(xg boost)

[9] Stacked Ensembles(2018,August,17)H2O.ai [Online] Available: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/stacked-ensembles.html

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay; “Scikit-learn: Machine Learning in Python” (Oct):2825−2830, 2011.

[11] P.Kanani, V.Kaul, K.Shah,” Hybrid PKDS in 4G using Securred DCC”, International Conference on Signal Propagation and Computer Technology (ICSPCT) 2014, pp- 323-328.

[12] M. Padole, P. Kanani, “Textimage Ciphering”, 2nd International Conference for Convergence in Technology (I2CT), 2017, pp - 926-930.

[13] Y. Doshi, A. Sangani, P. Kanani, M. Padole, “An Insight into CAPTCHA”, International Journal Of Advanced Studies In Computer Science And Engineering IJASCSE Volume 6, Issue 9, pp. 19-27. 2017.

[14] T. Reddy, J. Sanghvi, D. Vora, P. Kanani, “Wanderlust : A Personalised Travel Itinerary Recommender”, International Journal Of Engineering Development And Research IJEDR, Volume 6, issue 3, pp.78-83. 2018

[15] P. Kanani, K. Srivastava, J. Gandhi, D. Parekh, M. Gala, “**Obfuscation: maze of code”,** Proceedings of the 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA), IEEE (2017), pp.11-16

[16] Funda Güneş, Russ Wolfinger, and Pei-Yi Tan, “Stacked Ensemble Models for Improved Prediction Accuracy”, Paper SAS-2017

[17] Metrics to Evaluate Machine Learning algorithms in python (2018, August, 15) Machine Learning Mastery [Online] Available: https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/

[18] Root Mean Square deviation (2018, August,15) Wikipedia [Online] Available: https://en.wikipedia.org/wiki/Root-mean-square_deviation

[19] How To Interpret R-squared in Regression Analysis (2018, August,16) Blog By Jim Frost [Online] Available: http://statisticsbyjim.com/regression/interpret-r-squared-regression/

[20] Raschka, (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. Journal of Open Source Software, 3(24), 638, https://doi.org/10.21105/joss.00638

[21] The Potential of Machine Learning Real Estate Valuation Models (2018, August 12), Ershad Chagani, Cornell Blog [Online] Available: https://blog.realestate.cornell.edu/2018/03/28/machine-learning/