

Techniques to Improve the Performance of Supervised Learning Models

Manoj Sethi ¹, Lalit Kumar ², Shashi Bhushan Sharma ³ and Vikas ⁴

¹⁻⁴Department of Computer Science and Engineering, Delhi Technological University, Delhi, India

¹ Email: manojsethi@dce.ac.in, lalitkumar_2k17co169@dtu.ac.in, shashibhushansharma_2k17co319@dtu.ac.in, vikas_2k17co369@dtu.ac.in

Abstract—Supervised Learning can be defined as training a model with the data which includes the result itself. Many supervised learning algorithms have been found so far. There are a great number of supervised learning models. Each model performs differently and has its own merits and demerits. Many data preprocessing techniques have been found and hence the combination of various data preprocessing techniques can increase the performance of the present supervised learning models. Raw data contain a lot of noise so it cannot be fed to the learning models directly. It needs to be preprocessed using various data preprocessing techniques. The proposed work compares different combinations of various data preprocessing techniques. Comparison is done using various performance metrics and the combination of different data preprocessing is applied to different models. The work comprises categorical data handling, missing value treatment, feature scaling and feature extraction as the data preprocessing steps. The comparison gives an idea of which technique is better for which type of models. The California census 1990 data has been used for the study.

Index Terms— supervised learning, feature scaling, data preprocessing, feature extraction.

I. INTRODUCTION

The performance of a supervised learning model indicates how beneficial predictions are made by the trained model. Supervised Machine Learning first takes the instances along with the labels from the outer world to produce some hypotheses and then predict the labels for future instances. One cannot directly use raw data to train a model as it contains noise and irrelevant information[1]. The quality of data determines the success of a machine learning model and is of utmost importance. Data preprocessing is required to process the raw data and extract relevant information from that data which can be fed to the machine learning model[6].

The proposed approach uses some data preprocessing techniques to improve the performance of supervised learning models[6]. Two categorical data handling methods, two missing values treatment methods, two feature scaling methods and two feature extraction methods have been used. This work uses four different regression models to compare the results as Linear Regressor, Decision Tree Regressor, Random Forest Regressor, Support Vector Regressor and Gradient Boosting Regressor. A comparative study is done on which techniques are effective for the performance of supervised learning models[1].

The proposed work calculates various performance metrics for each regressor and then based on that compares each data preprocessing technique. Each model has a different capability. So, each model should perform differently than the other model for each data preprocessing technique[2]. Each regressor model has their own

advantages and disadvantages so they will perform differently on the same dataset, more specifically on the data which lie on the boundaries.

Hence this study provides the following advantages over the other studies performed earlier:

1. Comparative study of different combinations of various data preprocessing techniques.
2. Combination of data preprocessing techniques which are better for a particular type of model, are shown based on the values of different performance metrics. The rest of the paper shows the approach followed by comparative results. The last section contains the conclusion of the paper.

II. LITERATURE REVIEW

Various studies have been done regarding data preprocessing techniques. Supervised machine learning models have been introduced and compared [1]. It is required to make advances in the data preprocessing techniques to improve the performance of existing machine learning models. Data preprocessing includes missing value treatment, scaling of attributes, handling categorical attributes, extraction of relevant features, feature engineering, data compression, combination of attributes, etc [2]. One of the solutions for treating missing values is to either discard that attribute which has missing values or that instance which contains missing value. In the real world, there are so many missing values and discarding these values can result in a high loss of information. Batista Gustavo and Monard Maria Carolina treated the missing values using an imputation method based on k-nearest neighbour algorithm and showed that it outperforms the internal methods used by C4.5 and CN2 for treating missing values[3]. Stekhoven Daniel J. and Bühlmann Peter proposed an iterative imputation method MissForest based on random trees[4]. It works better than the conventional methods for nonlinear data. Karabulut Esra Mahsereci, Özel Selma Ayşe and brikçi Turgay showed that the performance of various models improved after the feature selection is applied to the dataset[5]. Feature selection technique takes those features into account which are important for making predictions. Significant improvement in the accuracy of Naive Bayes, Artificial Neural Network and J48 is observed after applying feature selection methods. Jayaram M.A., Karegowda Asha Gowda and Manjunath A.S. have selected the feature subset using the wrapper approach and used Genetic algorithm as a random search technique with various induction algorithms and the results are validated using classifiers[6]. He concludes that there is not a specific wrapper method which is better than the other but feature selection improves the classification accuracy. Chandrashekar Girish and Sahin Ferat used feature selection techniques and improved the performance for fault prediction analysis for Fault Mode Data[7]. More data are not always good for machine learning models. Lazar Cosmin et al. surveyed filter feature selection methods for valuable feature discovery in differentially expressed genes[8]. Jerez Jose M. et al. applied six imputation methods based on statistical methods such as mean imputation, hot-deck, machine learning techniques etc. and compared the results obtained from listwise deletion[9]. It is concluded that the machine learning imputation techniques performed better. Heaton Jeff has analysed feature engineering methods producing ten different features and showed that random forest and gradient boosting machines perform well for the same type of engineered features[10]. It also showed that ensemble methods perform better than the individual methods.

III. THE PROPOSED APPROACH

This study aims to compare various data preprocessing approaches which can increase the performance of supervised learning models. Some data preprocessing techniques used are : categorical data handling, missing value treatment, feature scaling and feature extraction[2]. Each data preprocessing technique uses two types of approach. Thus, a total 16 different approaches are obtained for data preprocessing. Using these techniques, data are preprocessed and then the results of various supervised learning models are compared using various performance metrics.

A. Categorical Data Handling

The data preprocessing starts with treating categorical data. Most of the machine learning libraries work on numeric data and cannot handle the categorical data. The categorical data is represented in some numeric form to make it usable for feeding the machine learning models. This work proposes two methods for treating categorical data: conversion into a number and dummy coding.

1. Conversion into a number converts the different levels of the categorical data of the specific attribute to unique numbers. The numbers range starting from zero to the levels of that categorical attribute.

2. Dummy coding is commonly used to convert the categorical data into numeric data. Dummy variables are created for each level present in the categorical attribute. The dummy variable is marked as one if that level corresponds to the given instance else that dummy variable is marked as zero.

Each of these methods converts the categorical data to numeric data. Using the first method, the dimension of the dataset remains the same as the old categorical attribute is replaced by the new numeric attribute but the second method increases the dimension of the dataset as the old categorical attribute is replaced by new dummy variables.

B. Missing Value Treatment

Missing data can deteriorate the performance of a machine learning model. Missing data can be treated by deleting the attribute or the instances having missing value. Missing value can also be treated by filling the missing value by some method. The proposed work treats the missing values by two ways: Mean/Mode/Median imputation and KNN Imputation[4].

1. Mean Imputation replaces the missing values by calculating the mean or median or mode of all the non-missing values of the feature.
2. KNN Imputation imputes the missing value by finding the specified nearest instances having non-missing values. The nearest instances are specified by the distance function. It can impute both qualitative and quantitative attributes. Attributes with multiple missing values can easily be treated. This algorithm is very time consuming.

C. Feature Scaling

Machine learning algorithms such as linear regression, etc. which uses gradient descent as an optimization approach. Features having the same range help the gradient descent to converge quickly at the minima. Distance based algorithms such as SVM, etc. also require scaling so that all features contribute equally. Two approaches are used here to scale the data: standardisation and normalisation.

1. Standardisation is a scaling technique in which the data is centred around the mean with a unit standard deviation. After obtaining the mean μ and the standard deviation σ of the feature values, the data X is standardised using the formula

$$X' = X - \mu \quad (1)$$

2. Normalisation is a scaling technique in which the data are shifted and scaled to a range of [0,1]. The minimum value is represented as x_{min} and the maximum value as x_{max} , then the data X is normalised using the formula

$$X' = \frac{(X - x_{min})}{(x_{max} - x_{min})} \quad (2)$$

Feature scaling is applied on the training data and not on the complete data or test data.

D. Feature Selection

The data consists of both, relevant and irrelevant information. Feature selection techniques select the features containing the most relevant information and discards those features which behave like noise[5]. This paper uses two feature selection methods: correlation based feature selection and recursive feature elimination.

1. Correlation defines the linear relationship between two variables. The correlation coefficient is calculated between the dependent and all other independent variables and then the specified number of features having highest correlation with the target feature are selected.
2. Recursive Feature Elimination removes the weakest feature until the specified number of features are reached. It tries to remove collinearity and dependencies which can exist in the model. Variable importance is determined by feature importance or correlation.

E. Prediction

After performing the proposed data preprocessing steps, to compare the results of our different data preprocessing approaches, five regressors namely Linear Regression, Decision Tree Regressor, Random Forest Regressor, Support Vector Regressor as well as ensemble models such as Gradient Boosting Regressor are selected. All regressors predict the continuous value of the target variable[10].

1. Linear Regression fits a line $y = mx + c$ to the given training data and predicts the test variable using this line. In case of multiple variables, a multidimensional plane fits to the training data and it is used for further prediction.

2. Decision Tree Regressor creates a tree where each node determines which type of instances fit that path. Each node uses an attribute to break it into further nodes. It predicts the value as the value corresponding to the leaf node where the instance reaches.
3. Random Forest Regressor is a collection of many decision tree regressors. The target value is predicted using all the decision trees and then averaged to get the final answer. It is also a kind of ensemble model.
4. Support Vector Regressor works the same as a support vector machine. It creates a hyperplane which maximises the margin and predicts the value for the target variable.
5. Gradient Boosting Regressor is an ensemble learning model. It adds the predictors sequentially by correcting its predecessor. When decision trees are used as predictors then it is called Gradient Boosting Regressor Trees

All the regression models predict a continuous numeric value of the target variable using the value of independent variables.

IV. RESULT AND DISCUSSION

A. Dataset and Data Preprocessing

This paper uses the California Housing Prices dataset from the StatLib repository to compare the results of various models using different combinations of data preprocessing methods[11]. This dataset is based on data from the 1990 California census.

To compare the results, data preprocessing is applied by performing categorical feature handling, missing value treatment, scaling and feature selection. Initially categorical features are handled using label encoding and dummy coding. The missing values in numeric data are then treated by mean imputation and knn imputation. The data are then scaled by standardisation and normalisation. The important attributes are extracted. Lastly, predictions are made by various supervised learning models.

B. Experimental Setup

All the models for this study have been implemented on a system with Windows 10 Home having 2.30 GHz Intel Core i5 processor and Python 3. The complete dataset has been splitted into ten parts to perform the hold one out validation and obtain more reliable results. For comparing the results of various models using different data preprocessing pipelines, five performance metrics namely mean squared error, mean absolute error, root mean squared error, r-square error and adjusted r square-error have been adopted.

C. Comparison of Data Preprocessing with Benchmark Regressors

This study compares the data preprocessing methods using various supervised learning models such as Linear Regression, Decision Tree Regressor, Random Forest Regressor, Support Vector Regressor as well as ensemble models such as Gradient Boosting Regressor.

Table 1,2,3,4 and 5 shows the results of various performance metrics applied to different algorithms using various combinations of data preprocessing techniques. It can be seen that the ensemble methods as Random Forest Regressor and Gradient Boosting Regressor perform better than the linear regression as well as decision tree regressor. Support Vector Regressor does not perform well over this dataset. But Gradient Boosting Regressor performs better than the Random Forest Regressor for every technique.

In case of Linear Regressor, combination of dummy coding, knn imputation, standardisation and correlation attribute selection works the best. In case of Decision Tree Regressor and Random Forest Regressor, combination of dummy coding, knn imputation, normalisation and correlation attribute selection works the best. In case of Support Vector Regressor, combination of dummy coding, mean imputation, normalisation and recursive feature elimination works the best. In case of Gradient Boosting Regressor, combination of label coding, knn imputation, standardisation and recursive feature elimination works the best.

So the comparison of various combinations over different models gives a fair idea about which data preprocessing techniques work for which type of model.

TABLE I: PERFORMANCE OF LINEAR REGRESSION

Technique	Mean Squared Error	Mean Absolute Error	Root mean squared error	R square error	Adjusted r square error
label_mean_std_corr	6374796738	59544.1084	78617.013	0.42276386	0.422596

label_mean_std_rfe	6394720219	59038.1103	79364.294	0.48672037	0.406739
label_mean_norm_corr	12735580803	87997.2882	111259.321	0.17090386	0.17892146
label_mean_norm_rfe	12956390284	80378.0392	118566.205	0.18574937	0.17595639
label_knn_std_corr	6374796738	59544.1084	78617.013	-0.422763860	-0.42317759
label_knn_std_rfe	6395762020	50283.2937	70365.365	-0.486292729	-0.40275089
label_knn_norm_corr	12735580803	87997.2882	111259.321	0.17090386	0.17066276
label_knn_norm_rfe	12947492028	80387.0376	115486.567	0.18464829	0.14049264
dummy_mean_std_corr	5863782984	49544.1084	68617.013	0.52276386	0.522596
dummy_mean_std_rfe	5803836120	40175.2930	60264.784	0.50275403	0.502796
dummy_mean_norm_corr	11739462820	77997.2882	101259.321	0.27090386	0.27892146
dummy_mean_norm_rfe	11947493740	725488.364	103547.494	0.20562938	0.27946259
dummy_knn_std_corr	5863782984	49544.1084	68617.013	-0.322763860	-0.32317759
dummy_knn_std_rfe	5803864828	40174.2784	69364.204	-0.302750572	-0.30576282
dummy_knn_norm_corr	11739462820	77997.2882	101259.321	0.27090386	0.27066276
dummy_knn_norm_rfe	11048482920	72964.8204	104769.706	0.20574374	0.27027484

TABLE II: PERFORMANCE OF DECISION TREE REGRESSOR

Technique	Mean Squared Error	Mean Absolute Error	Root mean squared error	R square error	Adjusted r square error
label_mean_std_corr	12339201277	77013.49	107476.443	-0.14253155	-0.07899738
label_mean_std_rfe	12750374048	77364.1944	104759.485	-0.17483947	-0.07027948
label_mean_norm_corr	11148537202	75451.0315	104415.264	0.032423615	0.036725373
label_mean_norm_rfe	11948363304	79365.3759	103894.485	0.030575927	0.030588595
label_knn_std_corr	11428723781	74597.2272	104807.292	0.085368362	0.085102391
label_knn_std_rfe	11963739203	79254.4759	107304.374	0.083694739	0.083936202
label_knn_norm_corr	11000346901	74999.5095	103695.604	0.040634007	0.040355026
label_knn_norm_rfe	11947495057	76374.2649	105303.374	0.049270282	0.049274947
dummy_mean_std_corr	11339205289	67013.49	97476.443	-0.04253155	-0.05899738
dummy_mean_std_rfe	11748463894	68364.2650	96204.384	-0.04959275	-0.05927492
dummy_mean_norm_corr	10148534209	65451.0315	94415.264	0.132423615	0.136725373
dummy_mean_norm_rfe	10746382937	69365.3850	96207.287	0.130583739	0.130559227
dummy_knn_std_corr	10428724135	64597.2272	94807.292	0.185368362	0.185102391
dummy_knn_std_rfe	10468339003	68254.5720	90364.384	0.189575938	0.180374937
dummy_knn_norm_corr	10000348430	64999.5095	93695.604	0.140634007	0.140355026
dummy_knn_norm_rfe	10369573648	67483.3884	92564.408	0.140686739	0.140472929

TABLE III: PERFORMANCE OF RANDOM FOREST REGRESSOR

Technique	Mean Squared Error	Mean Absolute Error	Root mean squared error	R square error	Adjusted r square error
label_mean_std_corr	7217492052	61608.234	82785.6461	0.327075547	0.32571246

label_mean_std_rfe	7203730227	60274.3943	86394.978	0.303840802	0.39750394
label_mean_norm_corr	6140788957	57345.6677	77035.005	-0.44506612	-0.44244165
label_mean_norm_rfe	6103783728	59387.2086	70265.985	-0.40284028	-0.48374739
label_knn_std_corr	7212618375	61634.3930	82509.522	-0.32512854	-0.32551388
label_knn_std_rfe	7293739479	60265.9254	84485.095	-0.39474284	-0.30484944
label_knn_norm_corr	6147323521	57445.4265	77081.524	-0.445107885	-0.44493084
label_knn_norm_rfe	6108023837	50275.9375	74385.474	-0.408840382	-0.44827494
dummy_mean_std_corr	6722992052	51608.234	72785.6461	0.427075547	0.42571246
dummy_mean_std_rfe	6793730273	56578.8562	72648.394	0.402840482	0.48364930
dummy_mean_norm_corr	5184088957	47345.6677	67035.005	-0.34506612	-0.34244165
dummy_mean_norm_rfe	5193830282	40987.3638	68376.305	-0.38475048	-0.39790303
dummy_knn_std_corr	6194018375	51634.3930	72509.522	-0.22512854	-0.22551388
dummy_knn_std_rfe	6193720284	59275.0275	76204.495	-0.20494928	-0.29474072
dummy_knn_norm_corr	5015823521	47445.4265	67081.524	-0.345107885	-0.34493084
dummy_knn_norm_rfe	5093749393	49867.3096	69376.384	-0.304830847	-0.37494839

TABLE IV: PERFORMANCE OF SUPPORT VECTOR REGRESSOR

Technique	Mean Squared Error	Mean Absolute Error	Root mean squared error	R square error	Adjusted r square error
label_mean_std_corr	14398637636	89667.7278	117299.214	-0.26571505	-0.26608311
label_mean_std_rfe	14769265025	85309.6397	118639.493	-0.26937103	-0.26917010
label_mean_norm_corr	14499708154	90041.4420	117733.478	0.27511489	0.274904101
label_mean_norm_rfe	14968539503	90365.7782	110375.798	0.27028457	0.270183820
label_knn_std_corr	14398637636	89667.7278	117299.214	0.265715046	0.265501518
label_knn_std_rfe	14970255036	80174.9479	110476.027	0.269207451	0.260193754
label_knn_norm_corr	14499708154	90042.4420	117733.478	0.275114895	0.2749041019
label_knn_norm_rfe	14067144067	97254.9408	110688.375	0.270294739	0.2701974529
dummy_mean_std_corr	13650157893	79667.7278	107299.214	-0.26571505	-0.26608311
dummy_mean_std_rfe	13960285304	70275.0578	100364.204	-0.26620674	-0.26927210
dummy_mean_norm_corr	13864190453	80041.4420	107733.478	0.27511489	0.274904101
dummy_mean_norm_rfe	13077249567	80285.9300	106394.047	0.27936505	0.270340370
dummy_knn_std_corr	13760165892	79667.7278	107299.214	0.265715046	0.265501518
dummy_knn_std_rfe	13086420867	70285.0275	108354.028	0.269364927	0.269103852
dummy_knn_norm_corr	13770158706	80042.4420	107733.478	0.275114895	0.2749041019
dummy_knn_norm_rfe	13029768257	80295.3839	109367.208	0.279372927	0.2708352910

TABLE V: PERFORMANCE OF GRADIENT BOOSTING REGRESSOR

Technique	Mean Squared Error	Mean Absolute Error	Root mean squared error	R square error	Adjusted r square error
label_mean_std_corr	5491005948	59667.7278	72754.428	0.503925384	0.503781128

label_mean_std_rfe	5380174837	58556.6167	71643.428	0.514936495	0.514936495
label_mean_norm_corr	6101700817	58117.1746	76286.799	-0.46293856	-0.46336398
label_mean_norm_rfe	6000600706	57006.0635	75175.799	-0.45184745	-0.45184745
label_knn_std_corr	5488026893	54186.7312	72738.095	-0.50412679	-0.504564188
label_knn_std_rfe	5377015782	53075.6201	71627.095	-0.50301568	-0.50301568
label_knn_norm_corr	6101696465	58117.4010	76287.0007	-0.462945118	-0.463370537
label_knn_norm_rfe	6000585354	57006.3000	75156.0006	-0.451834007	-0.451834007
dummy_mean_std_corr	4970105948	49667.7278	62754.428	0.603925384	0.603781128
dummy_mean_std_rfe	4860004837	48556.6167	61643.428	0.614936495	0.614936495
dummy_mean_norm_corr	5930100817	48117.1746	66286.799	-0.36293856	-0.36336398
dummy_mean_norm_rfe	5820000706	47006.0635	65175.688	-0.35182745	-0.35182745
dummy_knn_std_corr	4970126893	44186.7312	62738.095	-0.40412679	-0.404564188
dummy_knn_std_rfe	4860015782	43075.6201	61627.095	-0.40301568	-0.40301568
dummy_knn_norm_corr	5739196465	48117.4010	66287.0007	-0.362945118	-0.363370537
dummy_knn_norm_rfe	5628085354	47006.3000	65176.007	-0.351834007	-0.351834007

V. CONCLUSION

It is noted that most of the time of a programmer for training a model is spent on data preprocessing. Various combinations of data preprocessing techniques have been compared and different performance metrics have been evaluated to compare them. It can be seen that dummy coding for categorical data handling is better than label coding. Similarly, the use of standardisation for feature scaling gives better results most of the time. There is not any significant difference in the results after using knn or mean imputation and correlation based feature selection or recursive feature elimination.

REFERENCES

- [1] Caruana Rich and Niculescu-Mizil Alexandru, 2006, 'An empirical comparison of supervised learning algorithms', ICML '06: Proceedings of the 23rd international conference on Machine learning, pp. 161-168.
- [2] Kotsiantis S. B., Kanellopoulos D. and Pintelas P. E., 2006, 'Data Preprocessing for Supervised Learning', International journal of Computer Science, vol. 1, no. 1, pp. 111-117.
- [3] Batista Gustavo E. A. P. A. and Monard Maria Carolina, 2003, 'An Analysis of Four Missing Data Treatment Methods for Supervised Learning', Applied Artificial Intelligence, vol. 17, no. 5-6, pp. 519-533.
- [4] Stekhoven Daniel J. and Bühlmann Peter, 2012, 'MissForest—nonparametric missing value imputation for mixed-type data', Bioinformatics, vol. 28, no. 1, pp. 112-118.
- [5] Karabulut Esra Mahsereci, Özel Selma Ayşe and brikçi Turgay, 2012, 'A comparative study on the effect of feature selection on classification accuracy', Procedia Technology, vol. 1, pp. 323-327.
- [6] Jayaram M.A., Karegowda Asha Gowda and Manjunath A.S., 2010, 'Feature Subset Selection Problem using Wrapper Approach in Supervised Learning', International Journal of Computer Applications, vol. 1, no. 7, pp. 13-17.
- [7] Chandrashekar Girish and Sahin Ferat, 2013, 'A survey on feature selection methods', Computers & Electrical Engineering, vol. 40, no. 1, pp. 16-28.
- [8] Lazar Cosmin, Taminau Jonatan, Meganck Stijn, Steenhoff David, Coletta Alain, Molter Colin, Schaetzen Virginie de, Duque Robin, Bersini Hugues and Nowe Ann, 2012, 'A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis', IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 9, no. 4, pp. 1106-1119.
- [9] Jerez Jose M., Molina Ignacio, Garcia-Laencina Pedro J., Alba Emilio, Ribelles Nuria, Martín Miguel and Franco Leonardo, 2010, 'Missing data imputation using statistical and machine learning methods in a real breast cancer problem', Artificial Intelligence in Medicine, vol. 50, pp. 105-115.
- [10] Heaton Jeff, 2016, 'An empirical analysis of feature engineering for predictive modeling', Southeastcon, IEEE, pp. 1-6.
- [11] Pace, Kelley R. and Barry Ronald, 1997, 'Sparse Spatial Autoregressions', Statistics and Probability Letters, URL <https://github.com/ageron/handson-m>
- [12] Guyon I, Elisseeff A, 2003, 'An introduction to variable and feature selection', J Mach Learn Res, pp. 1165-1170.

- [13] Witten, I. H., & Frank, E. (2005). Data mining: Practical machine learning tools and techniques. San Francisco: Morgan Kaufmann. Second edition.
- [14] Changming Zhu, Daqi Gao, 2016, 'Influence of Data Preprocessing', Journal of Computing Science and Engineering, Vol. 10, No. 2, pp. 51-57.
- [15] Uma K, M. Hanumanthappa, 2017, 'Data Collection Methods and Data Preprocessing Techniques for Healthcare Data Using Data Mining', International Journal of Scientific & Engineering Research Volume 8, pp. 1131-1136.