Advanced Regression Techniques Based Housing Price Prediction Model

Preprint · February 2020

DOI: 10.13140/RG.2.2.18572.87684

1 author:

Sahar Abbasi

19 PUBLICATIONS **9** CITATIONS

SEE PROFILE

سیزدهمین کنفرانس بین المللی انجمن ایرانی تحقیق در عملیات
۳ و ۴ اردیبهشت ۱۳۹۹
13th International Conference of Iranian Operations Research Society
22-23 April 2020

Advanced Regression Techniques Based Housing Price Prediction Model

Sahar Abbasi
PhD student of Industrial Engineering, Payame Noor University, Tehran, Iran; Abbasi.iaun.ac@gmail.com

Abstract
In this paper, we are predicting the sale price of the houses using various machine learning algorithms. Housing sales price are determined by numerous factors such as material quality, living area square feet , Size of garage, location of the house number of bedrooms and so on. In particular, we focus on ۷۹ features that usually applicants consider them. We use the data of Kaggle website. We then use machine learning algorithms such as Randomforest, Ridge Regression, Lasso Regression and ElasticNet Regression to build the prediction model for houses to predict sale prices. Further, we have compared these algorithms based on RMSE parameter. Finally, to improve the accuracy of our model we used blend the results.
Keywords: Data science, Machine learning, Advanced Regression, Randomforest, Lasso Regression

۱. Introduction

Real estate is not only the key sector of the national economy, but also one of the citizen's major concerns. Due to the housing demands, people's attention to the housing price continues increasing. It is critical to provide accurate predictions of housing prices. Housing price is impacted by multiple factors ([۱], [۲]) including time and space, house ages, surrounding conditions, communities, transportation, etc. Existing prediction models are usually single predictor ones, i.e., a single forecasting model is applied for the prediction. The prediction accuracy of this type model is not satisfactory when datasets are noisy [۳]. In this paper we have considered various intrinsic parameters (such as number of bedrooms, living area and construction material) and also external parameters (such as location, proximity, upcoming projects, etc.). Then we have applied these parameter values to four different machine learning algorithms i.e. Randomforest, Ridge Regression, Lasso Regression and Elasticnet Regression and comparing them based on Root Mean Squared Error (RMSE). In addition to this we will also discuss the significance our approach and the methodology used.

All the experiments were run using the open-source software Python on a system with configurations ٤ GB RAM Mac iOS ١,٦ GHz Intel Core i٥.

۲. Related Work

In last two decades forecasting the property value has become an important field. A lot of researches have been done on Artificial Neural Networks. This has helped many researchers focusing on real estate problem to solve using neural networks. In [٤], the author has compared hedonic price model and ANN model that predict the house prices. Some researchers like that in [٥] have used classifiers to predict the property values. The author in research article [٥] has collected the data from

Multiple Listing Service (MLS), historical mortgages rates and public school ratings. Real Estate Data was obtained from Metropolitan Regional Information Systems (MRIS) database. The author extracted approximately ۱۵,۰۰۰ records from these three sources which included ۷۶ variables. Subsequently, t-test was used to select ۴۹ variables as a preliminary screening. Some researchers have focused on feature selection and feature extraction procedure. The author in article [۶] uses an open source data set of the housing sales in King County, USA. There are about ۲۰ explanatory variables. The author has compared various feature selection and feature extraction algorithms combined with Support Vector Regression. The author has collected approximately ۲۱,۰۰۰ observations in a time period of one year. The paper shows various data analysis performed on the data set. Feature Selection is the process of selecting a subset of variables from a given set of parameters either based on their importance or their frequency. However, feature extraction is the process of reducing the dimensionality of the data. Initial set of data is transformed into derived values which are equally informative and non-redundant. The three feature selection algorithms used are Recursive Feature Elimination (RFE), Lasso and Ridge and Random Forest Selector and the mean from each algorithm is calculated. Using feature selection, the author selects fifteen features out of twenty. The feature extraction algorithm used is Principal Component Analysis (PCA) and reduces the parameters from twenty to sixteen. However, the author found that both the techniques work equally well with the R squared value of ۰,۸۶.

Methodology represents a description about the framework that is undertaken. It consists of various

milestones that need to be achieved in order to fulfill the objective. We have undertaken different data mining and machine learning concepts. The following steps represents step-wise tasks that need to be completed:

- Data Collection
- Data preprocessing
- Handling missing values
- Handling Outliers
- Getting dummy categorical features
- Application of Algorithms
- Evaluating the models

۳. Methodology

۳,۱. Data Collection

The dataset used in this project was an open source dataset from KaggleInc [۷]. Our training data set included ۱۳۱٤ houses (i.e., observations) accompanied by ۸۰ attributes (i.e., features, variables, or predictors) and the sales price for each house. Our testing set included ۱٦۰٥ houses with the same ۷۹ attributes, but sales price was not included as this was our target variable. However out of these ۸۰ parameters only ۳۷ were chosen which are bound to affect the housing prices. Parameters such as Area in square meters, Overall quality which rates the overall condition and finishing of the house, Location, Year in which house was built, Numbers of Bedrooms and bathrooms, Garage area and number of cars that can fit in garage, swimming pool area, selling year of the house and Price at which house is sold. Selling price is a dependent variable on several other independent variables. After we've trained a model on *train.csv* data, we'll make predictions using the *test.csv* data. We'll simply recount our primary observations that will inform our feature engineering. We have ٤۳ categorical attributes which we will have to convert into dummy variables. Our first step was describe the futures in table ۱.

Table ۱. Describe the futures

future	description	future	description
• MSSubClass	The building class	• HeatingQC	Heating quality and condition
• MSZoning	The general zoning classification	• CentralAir	Central air conditioning
• LotFrontage	Linear feet of street connected to property	• Electrical	Electrical system
• LotArea	Lot size in square feet	• 1stFlrSF	First Floor square feet
• Street	Type of road access	• 2ndFlrSF	Second floor square feet
• Alley	Type of alley access	• LowQualFinSF	Low quality finished square feet (all floors)
• LotShape	General shape of property	• GrLivArea	Above grade (ground) living area square feet
• LandContour	Flatness of the property	• BsmtFullBath	Basement full bathrooms
• Utilities	Type of utilities available	• BsmtHalfBath	Basement half bathrooms
• LotConfig	Lot configuration	• FullBath	Full bathrooms above grade
• LandSlope	Slope of property	• HalfBath	Half baths above grade
• Neighborhood	Physical locations within Ames city limits	• Bedroom	Number of bedrooms above basement level
• Condition1	Proximity to main road or railroad	• Kitchen	Number of kitchens
• Condition2	Proximity to main road or railroad (if a second is present)	• KitchenQual	Kitchen quality
• BldgType	Type of dwelling	• TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
• HouseStyle	Style of dwelling	• Functional	Home functionality rating
• OverallQual	Overall material and finish quality	• Fireplaces	Number of fireplaces
• OverallCond	Overall condition rating	• FireplaceQu	Fireplace quality
• YearBuilt	Original construction date	• GarageType	Garage location
• YearRemodAdd	Remodel date	• GarageYrBlt	Year garage was built
• RoofStyle	Type of roof	• GarageFinish	Interior finish of the garage
• RoofMatl	Roof material	• GarageCars	Size of garage in car capacity
• Exterior1st	Exterior covering on house	• GarageArea	Size of garage in square feet
• Exterior2nd	Exterior covering on house (if more than one material)	• GarageQual	Garage quality
• MasVnrType	Masonry veneer type	• GarageCond	Garage condition
• MasVnrArea	Masonry veneer area in square feet	• PavedDrive	Paved driveway
• ExterQual	Exterior material quality	• WoodDeckSF	Wood deck area in square feet
• ExterCond	Present condition of the material on the exterior	• OpenPorchSF	Open porch area in square feet
• Foundation	Type of foundation	• EnclosedPorch	Enclosed porch area in square feet
• BsmtQual	Height of the basement	• 3SsnPorch	Three season porch area in square feet
• BsmtCond	General condition of the basement	• ScreenPorch	Screen porch area in square feet
• BsmtExposure	Walkout or garden level basement walls	• PoolArea	Pool area in square feet
• BsmtFinType1	Quality of basement finished area	• PoolQC	Pool quality
• BsmtFinSF1	Type 1 finished square feet	• Fence	Fence quality
• BsmtFinType2	Quality of second finished area (if present)	• MiscFeature	Miscellaneous feature not covered in other categories
• BsmtFinSF2	Type 2 finished square feet	• MiscVal	$Value of miscellaneous feature
• BsmtUnfSF	Unfinished square feet of basement area	• MoSold	Month Sold
• TotalBsmtSF	Total square feet of basement area	• YrSold	Year Sold
• Heating	Type of heating	• SaleType	Type of sale
		• SaleCondition	Condition of sale

۳,۲. Data preprocessing

Data preprocessing is a process of transforming the raw, complex data into systematic understandable knowledge. It involves the process of finding out missing and redundant data in the dataset. Entire dataset is checked for *NaN* and whichever observation consists of NaN will be deleted. Thus, this brings uniformity in the dataset. Before applying any model to our dataset, we need to find out characteristics of our dataset. Thus, we need to analyze our dataset and study the different parameters and relationship between these parameters. We can also find out the outliers present in our dataset. Outliers occur due to some kind of experimental errors and they need to be excluded from the dataset. First, We used Series.describe() to get more information.

count	۱۳۱٤,۰۰۰۰۰۰
mean	۱۸۱۳۰۷,٥۳۱۲۰۲
std	۷۸٤۰۲,٤٤۹۰۹۳
min	۳٤۹۰۰,۰۰۰۰۰۰
۲٥٪	۱۳۰۰۰۰,۰۰۰۰۰۰
٥۰٪	۱٦۳۲٥۰,۰۰۰۰۰۰
۷٥٪	۲۱٤۳۷٥,۰۰۰۰۰۰
max	۷٤٥۰۰۰,۰۰۰۰۰۰

Name: SalePrice, dtype: float٦٤

The results show that the average sale price of a house in our dataset is close to $۱۸۱,۰۰۰, with most of the values falling within the $۱۳۰,۰۰۰ to $۲۱٥,۰۰۰ range.
Then we do some plotting during the exploration stage of our project that allows us to visualize the distribution of the data, check for outliers, and see other patterns that we might miss otherwise.so, we check for skewness, which is a measure of the shape of the distribution of values. plot a histogram of SalePrice and Theoretical quintiles is shown in figure۱ and figure۲ respectively.

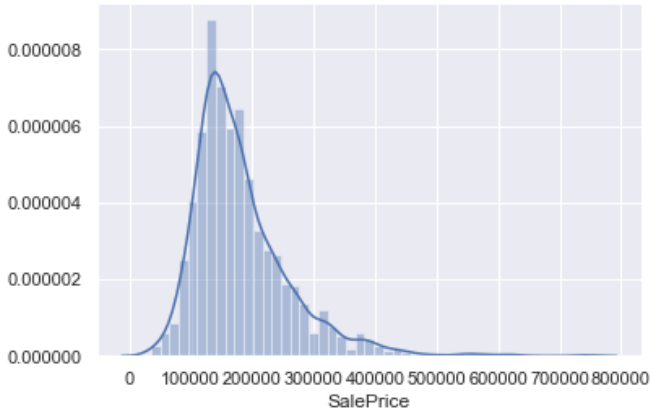

Fig.۱: histogram of SalePrice

We see that the target variable SalePrice has a right-skewed distribution. We'll need to log transform this variable so that it becomes normally distributed. A normally distributed (or close to normal) target variable helps in better modeling the relationship between target and

independent variables. In addition, linear algorithms assume constant variance in the error term. Alternatively, we can also confirm this skewed behavior using the skewness metric.

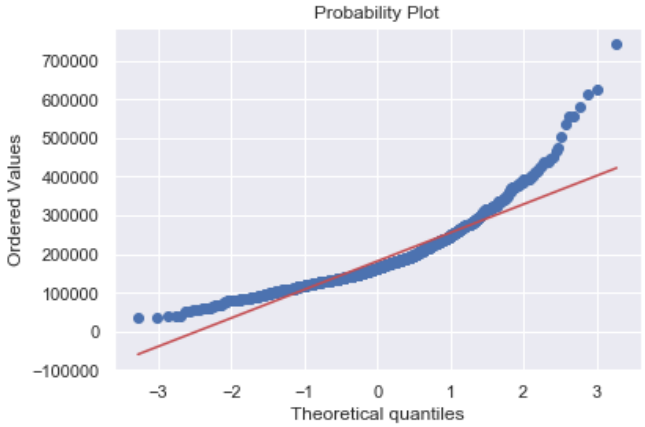

Fig۲: Theoretical quintiles

Then we use np.log() to transform train.SalePric and calculate the skewness a second time, as well as re-plot the data. Importantly, the predictions generated by the final model will also be log-transformed, so we need to convert these predictions back to their original form later. np.log() will transform the variable, and np.exp() will reverse the transformation. The skew in figure۳ seems now corrected and the data appears more normally distributed.results are shown in figure ۳,٤.

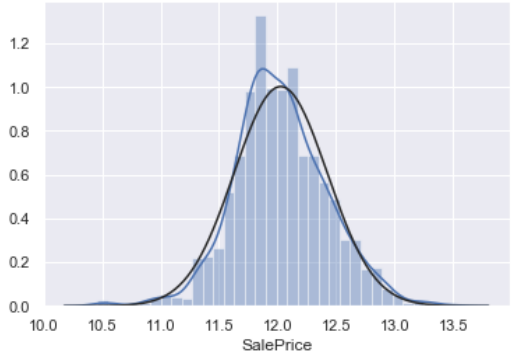

Fig۳: normally distributed Sale price histogram

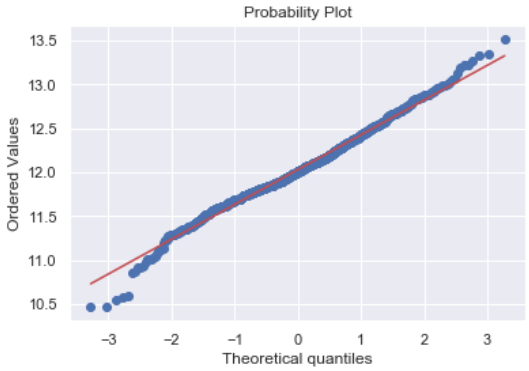

Fig٤: Normally Theoretical quintiles

As you see, log transformation of the target variable has helped us fixing its skewed distribution and the new distribution looks closer to normal. Since we have ۸۰ variables, visualizing one by one wouldn't be

an astute approach. Instead, we'll look at some variables based on their correlation with the target variable. However,

there's a way to plot all variables at once, and we'll look at it as well. Moving forward, we'll separate numeric and categorical variables and explore this data from a different angle. A correlation number gives the degree of association between two variables.

The correlation number exists between +۱ to -۱. A positive number represents a positive correlation between two variables and vice versa. However, if the correlation number is zero, it shows that there is no correlation between two variables and they are independent of each other. Correlation Matrix gives an in depth idea about correlation among various parameters. We have plotted a correlation matrix for ۱۵ selected parameters in the following figure ۵.

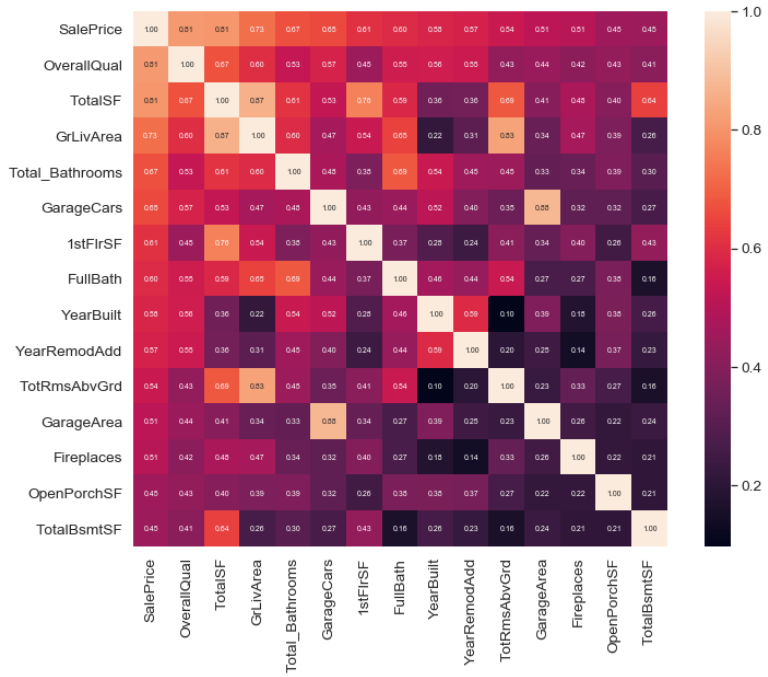

Figure ۵: Correlation Matrix

۳,۳. Handling Outliers

In the dataset, there are many homes with zero value for Garage Area, indicating that they don't have a garage. We'll transform other features later to reflect this assumption. There are a few outliers as well. Outliers can affect a regression model by pulling our estimated regression line further away from the true population regression line. So, we remove or replace those observations from our data. There are number of values that seem rather strange. In both training and test sets, we have several garages that were built as many as ۲۰ years earlier than their houses and in the training set we have a garage from the future – the record claims that it was built in ۲۲۰۷! Clearly something has gone wrong with these entries and – if we have some means to do so – we would ideally replace them with corrected values. If this is not possible, however, we can proceed to treat them as if they were missing.

۳,٤. Handling missing values

In many instances, Python interpret as a missing value is actually a class of the variable. For example, Pool Quality is comprised of ۵ classes: Excellent, Good, Fair, Typical, and NA. The NA class describes houses that do not have a pool, but our coding languages interpret houses of NA class as a missing value instead of a class

of the Pool Quality variable. Our solution was to impute most of the NA/NaN values to a value of "None." We create a DataFrame to view the top null columns. Chaining together the train.isnull().sum() methods, we return a Series of the counts of the null values in each column. The results is shown in figure ٦.

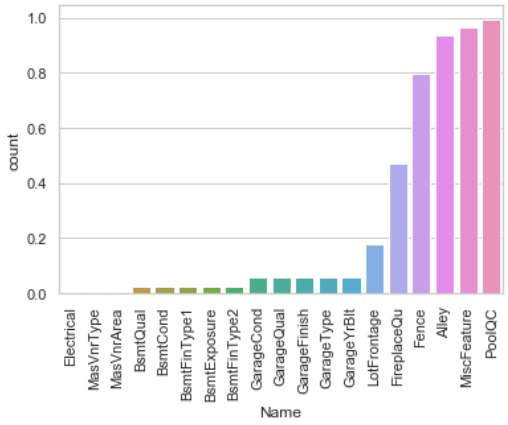

Fig٦.null columns

While we found that most NA/NaN values corresponded to an actual class for different variables, some NA/NaN values actually represented missing data. For example, we find that three houses with NA/NaN values for Pool Quality, also have a non-zero value for the variable, Pool Area (square footage of pool). These three houses likely have a pool, but its quality was not assessed or input into the data set.Our solution was to first calculate median Pool Area for each class of Pool Quality, then impute the missing Pool Quality classes based on how close that house's Pool Area was to the mean Pool Areas for each Pool Quality class. Some variables had a moderate amount of missingness. For example, about ١٧٪ of the houses were missing the continuous variable, Lot Frontage, the linear feet of street connected to the property. Intuitively, attributes related to the size of a house are likely important factors regarding the price of the house. Therefore, dropping these variables seems ill-advised. Our solution was based on the assumption that houses in the same neighborhood likely have similar features. Thus, we imputed the missing Lot Frontage values based on the median Lot Frontage for the neighborhood in which the house with missing value was located.

Most variables have some intuitive relationship to other variables, and imputation can be based on these related features. But some missing values are found in variables with no apparent relation to others. For example, the Electrical variable, which describes the electrical system, was missing for a single observation. Our solution was to simply find the most common class for this categorical variable and impute for this missing value.

There are two types of categorical features: ordinal, where there is an inherent order to the classes (e.g., Excellent is greater than Good, which is greater than Fair), and nominal, where there is no obvious order (e.g., red, green, and blue).Our solution for ordinal variables was to simply assign the classes a number corresponding to their relative ranks. For example, Kitchen Quality has five classes: Excellent, Good, Typical, Fair, and Poor, which we encoded (i.e., converted) to the numbers ٥, ٤, ٣, ٢, and ١, respectively. The ranking of nominal categories is not appropriate as there is no actual rank among the classes. Our solution was to one-hot encode these variables, which creates a new variable for each class with values of zero (not present) or one (present).

٣,٥. Feature Engineering

Feature (variable or predictor) engineering is one of the most important steps in model creation. Often there is valuable information "hidden" in the predictors that is only revealed when manipulating these features in some way. Below are just some examples of the features we created:

- Remodeled (categorical): Yes or No if Year Built is different from Year Remodeled; If the year the house was remodeled is different from the year it was built, the remodeling likely increases property value
- Seasonality (categorical): Combined Month Sold with Year Sold; While more houses were sold during summer months, this likely varies across years, especially during the time period these houses were sold, which coincides with the housing crash (۲۰۰۶-۲۰۱۰)
- New House (categorical): Yes or No if Year Sold is equal to Year Built; If a house was sold the same year it was built, we might expect it was in high demand and might have a higher Sale Price
- Total Area (continuous): Sum of all variables that describe the area of different sections of a house; There are many variables that pertain to the square footage of different aspects of each house, we might expect that the total square footage has a strong influence on Sale Price.

۳,٦. Getting dummy categorical features

Since the majority of available machine learning algorithms can only take numbers (floats or integers) as inputs, we must encode these features numerically if we are to use them in our models. The way we go about this will vary depending on the nature of the feature and the model we decide to use.
Non-numerical variables tend to come in two flavours: ordinal and categorical. Ordinal variables – such as OverallQual or LotShape – have an intrinsic order to them, while purely categorical variables – such as Neighborhood or Foundation – do not. The main approach to treating categorical variables is known as dummy or one-hot encoding. We apply dummy encoding to this as follows:

```
train =pd.get_dummies(train)
print(train.shape)
test = pd.get_dummies(test)
print(test.shape)
(۱۳۱٤, ۲۲٦)
(۱٦۰٥, ۳۲۲)
```

٤. Models and Results

Once the data is clean and we have gained insights about the dataset, we can apply an appropriate machine learning model that fits our dataset. We have selected four algorithms to predict the dependent variable in our dataset. The algorithms that we have selected are basically used as classifiers but we are training them to predict the continuous values. The four algorithms are LASSO Regression, Elastic Net Regression, Ridge Regression and RandomForest Regression. These algorithms were implemented with the help of python's SciKit-learn Library [۸]. The predicted outputs obtained from these algorithms were saved in comma separated value file. This file was generated by the code at run time. This model may be very sensitive to outliers. So we need to make it more robust on them. For that we use the sklearn'sRobustscaler() method on pipeline.

```
lasso = make_pipeline(RobustScaler(), LassoCV(max_iter= ۱e۷, alphas=alphas ۲,random_state= ٤۲, cv=kfolds))
elasticnet = make_pipeline(RobustScaler(), ElasticNetCV(max_iter= ۱e۷, alphas=e_alphas, cv=kfolds, l ۱_ratio=e_l ۱ratio))
ridge = make_pipeline(RobustScaler(), RidgeCV(alphas=alphas_alt, cv=kfolds))
```

```
randomforest=RandomForestRegressor(n_estimators= ٦٠٠,n_jobs=,oob_score=True,min_samples_leaf
= ٢,max_features= ٠,٥)
models = {'Ridge': ridge,
        'Lasso': lasso,
        'ElasticNet': elasticnet,
        'RandomForest': randomforest}
predictions = {}
scores = {}
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size= ٠,٢)
for name, model in models.items():
model.fit(X_train, y_train)
predictions[name] = np.expm ١(model.predict(X_train))
score = cv_rmse(model, X=X)
scores[name] = (score.mean(), score.std())
score = cv_rmse(ridge)
print("Ridge score: {:. ٤f} ({:. ٤f})\n".format(score.mean(), score.std()))
Ridge score: ٠,٠١٢٧ (٠,٠٠٢٤)
score = cv_rmse(lasso)
print("Lasso score: {:. ٤f} ({:. ٤f})\n".format(score.mean(), score.std()))
Lasso score: ٠,٠١٢٤ (٠,٠٠٢٣)
score = cv_rmse(elasticnet)
print("ElasticNet score: {:. ٤f} ({:. ٤f})\n".format(score.mean(), score.std()))
ElasticNet score: ٠,٠١٢٥ (٠,٠٠٢٢)
score = cv_rmse(randomforest)
print("RandomForest: {:. ٤f} ({:. ٤f})".format(score.mean(), score.std()))
      RandomForest: ٠,٠١٤٧ (٠,٠٠٢٤)
```

Table ٢. Results

Model	RMSE
randomforest	٠,٠١٤٧
elasticnet	٠,٠١٢٥
lasso	٠,٠١٢٤
ridge	٠,٠١٢٧

From the table ٢ we find that Lasso has least RMSE values. We begin with this simple approach of averaging base models. We build a new class to extend scikit-learn with our model and also to laverage encapsulation and code reuse.

```
def blend_models_predict(X):
return (٠,٢٥ *randomforest.predict(X) + \
        (٠,٢٥ *elasticnet.predict(X)) + \
        (٠,٢٥ * lasso.predict(X)) + \
        (٠,٢٥ * ridge.predict(X)) )
print('RMSLE score on train data:')
print(rmsle(y, blend_models_predict(X)))
RMSLE score on train data:
٠,٠١٠٢٧٠٠١٥٧٣٥٧٤٣٣١٢
```

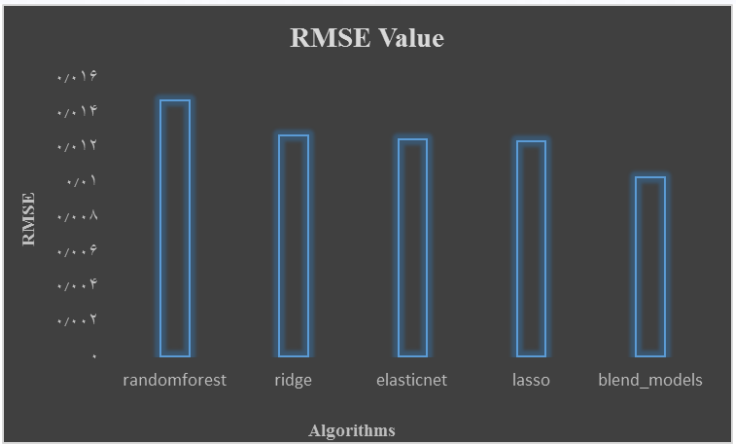

Fig۷.Algorithms vs RMSE

۵. Conclusion

In this research paper, we have used machine learning algorithms to predict the house prices. We have mentioned the step by step procedure to analyze the dataset and finding the correlation between the parameters. Thus we can select the parameters which are not correlated to each other and are independent in nature. These feature set were then given as an input to four algorithms and a csv file was generated consisting of predicted house prices. Hence we calculated the performance of each model using RMSE metric and compared them. We found that Blend_models and Lasso Regression overfits our dataset and give the least RMSE of ۰,۱۰۳ and ۰,۱۲٤ respectively. For future work, we recommend that working on large dataset would yield a better and real picture about the model. We have undertaken only few Machine Learning algorithms that are actually classifiers but we need to train many other classifiers and understand their predicting behavior for continuous values too. By improving the error values this research work can be useful for development of applications for various respective cities.

References

[۱]. Park B. and Bae J K. Using machine learning algorithms for housing price prediction. Expert Systems With Applications, ٤٢: ٢٩٢٨ –٢٩٣٤, ٢٠١٥.
[۲]. Glaeser E L and Nathanson C G. An extrapolative model of house price dynamics. Journal of Financial Economics, ٢٠١٧.
[۳]. Rajan U, Seru A, and Vig V. The failure of models that predict failure: Distance, incentives, and defaults. Journal of Financial Economics, ١١٥: ٢٣٧ –٢٦٠, ٢٠١٥.
[٤]. Limsombunchai, Visit. "House price prediction: hedonic price model vs. artificial neural network."New Zealand Agricultural and Resource Economics Society Conference. ٢٠٠٤.
[٥]. Park, Byeonghwa, and Jae Kwon Bae. "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data."Expert Systems with Applications ٤٢,٦ (٢٠١٥): ٢٩٢٨-٢٩٣٤.
[٦] Wu, Jiao Yang. "Housing Price prediction Using Support Vector Regression." (٢٠١٧).
[٧]. https://www.kaggle.com/ohmets/feature-selection-forregression/data
[٨]. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research ١٢.Oct (٢٠١١): ٢٨٢٥-٢٨٣٠