House Price Prediction Using Machine Learning Techniques

1st Mete DURLU
Department of Computer Engineering
Baskent University
Ankara, TURKEY
metedurlu@gmail.com

2nd Duygu Dede SENER
Department of Computer Engineering
Baskent University
Ankara, TURKEY
ddede@baskent.edu.tr

***Abstract*—Forecasting provides an insight about future trends in various research fields. House price forecasting is one of the most important topics of real estate. Machine learning methods have gained importance to predict the price of a house with the rapid growth of real estate market. In this study, a useful house price prediction model is aimed to propose with using machine learning algorithms. Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Dense Neural Networks (DNN) are used techniques in our model. Experimental results demonstrate that DNN based on accuracy has better performance rather than the other techniques in house price prediction.**

***Index Terms*—Forecasting, house price prediction, regression, machine learning, Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Dense Neural Networks (DNN).**

I. Introduction

With development of big data analysis, many fields of study have emerged over the last decade. Even if the concept is relatively new, methods behind the process are rather well established and been widely used for a long period of time. With such tools enabling us to reveal hidden patterns from the depths of big data, it has drawn attention from nearly all fields of study. Since machine learning methods have drawn so much attention, it provided the opportunity for a rapid establishment period for the market. Nowadays every business is trying to integrate some kind of machine learning or neural network in their work flow so they can benefit from the vast upside of accumulated data. For instance Machine learning(ML) and artificial neural network(ANN) methods have shown great upside in field of medical studies since it decreased the amount of effort in researches, while enabling experts to analyse test data much faster. In field of economics ML and ANN provided valuable predictions for the investors and business owners because these methods accounted patterns and rules which are unheard of or indescribable for even a data scientist. When these examples are taken in mind, ML and ANN methods show great potential for our problem; house price prediction. Accurate prediction of house prices has been a great challenge for real estate investors and for city planners. Since 1999 rules and states affecting house prices were trying to be explained but back then it was uncommon to use ML and especially ANN methods for such tasks. Later in 2004 studies with Artificial Neural Networks yielded promising results since ANNs can represent non-linear models for prediction. Since then considerable efforts has been made for improving ML and ANN methods so they have become more ideal solutions for these kinds of tasks. In this work we have used Ames Housing Dataset as training and testing data while performing several ML and ANN methods in order to predict the Sale Price of our samples. Among selected ML methods, a Naive Bayes approach has been taken but it has been observed that it lacked the sophistication. Another approach was to use the notorious K-Nearest Neighbors(KNN) algorithm. It enabled a robust prediction while the whole dataset was present and viable for estimation but the issue was that it was unsuitable for a realworld dataset since it is impossible to access all data while inference takes place. Our last ML method SVM uses kernels in order to present non-linear models which is the most powerful side of SVM but it can also be a downside since these kernels bring a lot of hyperparameters for tuning. These hyperparameters effect the model in such way that, tuning them can require considerable amount of time and resources. Our last approach and proposed solution was our artificial neural network with a prediction confidence of 86%. With the ability of handling features only using a single error function enables model tuning tuning lesser hyperparameters and still produces high success rates. Later in our work it has been discussed how these models compare with each other in a more detailed view and why we have chosen ANNs for House Price Prediction.

II. Related Works

Assessment of future real estate prices has been an important task since a lot of work has been dedicated for this specific issue. In 1975 HPI [1](housing price index) has been proposed as a price prediction method which is a weighted, repeat-sales index. HPI was limited with selected mortgage transactions. Several more indexes were also made such as SP(Case-Shiller) [2] index or FNC Residential Pricing Index. In 2014 UK decided to use a single official index, therefore came UK HPI [3]. It was proposed in order to prevent polarization on different index publications for UK. Merging all statistical data, a hedonic regression method has been used to determine HPI for UK. One of the most important issues were choosing the time period of data. This emphasizes the importance of data refinement. Hedonic regression, being a complex and statistically heavy method, can be hard to interpret and understand therefore we have shifted our attention to other machine learning methods.

A related solution came when Sifei Lu et al at 2017 [5] proposed a hybrid regression technique for house price prediction. By utilizing Ames Housing Dataset, several regression methods were used to experiment on data. Ridge, Lasso and GradientBoost regression methods has been applied and several issues were addressed. These issues include anomalies in data, overfitting of models and importance of feature engineering. Furthermore results indicate that regression methods can yield accurate results when parameters are tuned.

ANNs were also proposed [4] as a solution for regression problems and proved to be at least comparable with machine learning regression methods. Methods such as linear regression, polynomial regression, SVMs, Regression trees and ANNs have been compared while linear regression has been taken as baseline. It has been understood that proposed ANN with two hidden layers performed significantly worse than any other regression method mentioned. This issue grabbed our attention towards building a deeper neural network with more variables. With more variables and neurons, constructed ANN could perform better.

From previous works about house price prediction, three main issues have been grasped; data should be subjected to some form of pre-processing(such as anomaly removal) beforehand, a considerable effort goes into hyper-parameter tuning and artificial neural networks should be designed according to specific problem at hand.

III. Methods

In this section we are going to explain used methods for pre-processing, regression and classification.

A. Dealing With Missing Data

After careful inspection of data-set at hand it has been observed that missing values in data are actually correspond to missing features in houses. If an example needs to be given; when a house has a garage, ”garage size in square feet” has a positive numerical value, if there is no garage present then related feature gets a ”Nan” value. For numerical features ”Nan” values were replaced with average of feature values. For categorical features problem has been handled by replacing ”Nan” values with ”none”, this enabled us to include those samples to categorical encoding accurately.

B. Categorical Encoding

Since the data-set included 38 categorical features, we had to make use of them. Features were consisting of string values and represented predefined types. For example; MSZoning (general zoning classification) had values such as;

- A (Agriculture)
- C (Commercial)
- FV (Floating Village Residential)
- I (Industrial)
- RH (Residential High Density)
- RL (Residential Low Density)
- RP (Residential Low Density Park)
- RM (Residential Medium Density)

on such a case it was logical to use categorical encoding and represent these types as numbers.

C. Normalization

The goal of normalization is to set the data in a linear form. This operation improves the performance and prevents the unbalanced distribution of feature values. Minimum-maximum (min-max) normalization (1) was performed on data-set before applying each method. Let X be the value of a feature, X_{min} is the minimum value and X_{max} is the maximum value of that feature.

$$X^{'} = (X - X_{min})/(X_{max} - X_{min}) \tag{1}$$

D. Feature Correlation Matrix

Feature Correlation is basically the strength of relationship between features. Correlation can be calculated between each two feature independently. These relations makes possible to locate semantically identical features and most importantly gives information about which features affect the target the most.

Pearson Correlation Formula:

$$r(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

- n = sample size
- i = sample index
- $\bar{x}$ = average of feature samples
- 1 indicates a strong positive relationship between x,y.
- -1 indicates a strong negative relationship between x,y.

Feature correlation matrix was used to obtain the correlation values between each feature and our target (salePrice). Our approach was to keep features which have correlation values above a certain predefined threshold value. For this task we have taken the absolute values of each correlation value and determined a threshold. Correlation coefficient values below 0.2 are considered to be weak; 0.3-0.7 are moderate; values bigger than 0.7 are strong according to statistical studies [9]. Our threshold value (”0.2”) enabled us to reduce 81 features into 22 while maintaining minimum significance. Also the lesser feature count helped us with both data anomalies and also with shortening model training time.

E. Artificial Class Construction

When dealing with continuous data, pattern recognition methods are limited with regression techniques. Our data-set has continuous data as target values. This situation demands creative workarounds such as ”Artificial Class Construction” as our terms. In order to keep the data distribution as original we have used the standard deviation as the denominator. To keep the amount of classes less we have doubled the standard deviation before dividing the target values. Using equation below we have managed to get 5 distinct classes while managing to keep the distributions of classes same as the original data-set.

$$r = (\frac{x}{2\sigma}) \tag{3}$$

r = class label
x = sample's sale price
σ = standard deviation of whole data-set

F. Classification Methods

Classification is actually a method for predicting class values by using feature values of a samples through certain processes or equations. It can be achieved through either supervised or unsupervised methods. The most common methods are Naive Bayes, K-Nearest Neighbors and Support Vector Machine. These methods are all supervised methods where the number of classes and class labels for each training sample is known as prior knowledge. All mentioned methods have been tried on data-set at hand with artificially crafted classes. Conclusions can be seen on results part with all evaluation applied.

- **Naive Bayes**
  Naive Bayes method is the most common and easy to implement supervised classification method. Although it is simple, it can produce relatively good results in some real-life cases. In this regard it has became the first method to try before anything else.

  $$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{4}$$

  $$P(c|X) = \prod_{i=1}^{n} P(x_i|c) \tag{5}$$

  P(c|x) is the posterior probability of class given feature x is present. P(c) is the prior probability of class. P(x|c) is the likelihood which is the features probability of occurrence on given class. P(x) is the prior probability of feature.
  The main reason, it is not superior to the more complex methods, is the fact that Naive Bayes makes the assumption of features being independent from each other. Unfortunately this assumption is almost always wrong for our data-set. Due to this simple fact Naive Bayes yielded sub-optimal results for predictions outside of training-set. Observations on training-set showed 99% accuracy while circa 30% on test data.

- **K-Nearest Neighbors Clasiffier (KNN)**
  K-Nearest Neighbors is a really powerful method with many applications. These applications include both classification and regression. It is simple, easy to implement but has a tendency of getting too computationally expensive due to its nature. It is a supervised method relying on training sets class labels to classify new data. Basically it computes distance between each training sample and new input sample. Then uses the least distant "K" neighbors in order to decide on the class information. For example, if "K" equals three and then there are three votes to decide on new samples class. So three nearest neighbors could have class labels of one, one and zero. Then algorithm decides that new sample should have a class label of one. There is no need for hyper-parameters except "K" and there is no training but it becomes computationally expensive to calculate distances as the dimension and sample size increases. Most commonly euclidean distance is used to measure the distance between samples.

  $$d(x,y) = \sum_{i=1}^{n} \sqrt{(x_i - y_i)^2} \tag{6}$$

  d = distance between sample x and sample y
  x_i = i^{th} feature of sample x
  y_i = i^{th} feature of sample y
  n = number of features (dimenion of feature space)

- **Support Vector Machine Classifier (SVC)**
  Support Vector Machine is a supervised machine learning method which is a powerful and efficient tool. It can be used for both classification and regression problems. SVMs main goal is finding a hyperplane that best divides a data-set to two different classes multiple times(as many times needed to match number of classes). Support vectors are the data points nearest to the hyperplane, these point help define the hyperplane so all computations are done through these points. This hyperplane creates an area of margin which divides two classes apart. Error function here is designed so that the margin becomes larger as error decreases. If there is no clearly dividing hyperplane then the whole feature space is transformed into a new higher dimension feature space. This is known as kernelling. SVMs produce accurate results on clean data-sets with small to medium sample size. When dealing with larger data-sets however computational costs can be too much to handle and also it is highly sensitive to the noisy nature of large data-sets.

  $$\sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \tag{7}$$

  y = class based constant always ≥ 1
  α = Function coeffecent
  K = Kernel Function
  x = Support Vector
  b = Independent Term or Intercept

G. Regression Methods

Regression analysis is a predictive technique which investigates the relationship between target feature and other given features in feature space. This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.

- **K-Nearest Neighbors Regression**
  K-Nearest Neighbors [6] algorithm has already been mentioned as a Classification algorithm but it can also be used as a regression method. Logic behind it has many similarities with Interpolation methods. Main method of KNN regression is to calculate the average of the numerical target of the K nearest neighbors. KNN regression uses the same distance functions as KNN classification.

$$d(x,y) = \sum_{i=1}^{n} \sqrt{(x_i - y_i)^2} \tag{8}$$

  d = distance between sample x and sample y
  x_i = i^{th} feature of sample x
  y_i = i^{th} feature of sample y
  n = number of features (dimension of feature space)

- **Support Vector Machine Regression(SVR)**
  It has been already mentioned that SMVs are good at both regression and classification. Reminding, SVMs main goal is finding a hyperplane that best divides a data-set to two different classes for classification. On regression [7] this hyperplane actually tries to mimic data-sets behaviour and error function tries to minimize points outside of the determined margin. Basically SVR is trying to decide a decision boundary at 'C' distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line.

$$Prediction = \sum_{i \in SV} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{9}$$

  α,α^* = Function coefficients, bound to given distance 'C'
  K = Kernel Function
  x = Support Vector
  b = Independent Term or Intercept

- **Dense Neural Network Regression**
  Over the last decade Neural Networks have became the primal solution to many problems including classification and regression in data-sets of all sizes. A Neural Network usually described as a blackbox taking input and producing output but actually they can be simply explained as long functions with adjusted weights. The smallest units in the Neural Networks are neurons. And the whole structure consists of layers made from neurons. Each neuron in a layer receives an input from all the neurons present in the previous layer. Before an input is used in a neuron it will be multiplied by the weight value adjusted by the optimizer in Neural Network. Optimizer as in its name optimizes the weight values so that the final output becomes similar to the desired value. On each iteration, output will be put in an error function and according to the error optimizer will adjust the weights on Neural Network. It could be imagined as a hyperplane of errors calculated by each output pushed out. The goal is to find the global minimum of the hyperplane by adjusting weights. There are a lot of equations to choose from in order to optimize the weights and calculate the error. The problem on hand should be considered before selecting these equations. In our situation:
  Optimizer: RMSprop

$$\begin{aligned} E[g^2]_t &= 0.9E[g^2]_{t-1} + 0.1g_t^2 \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \end{aligned} \tag{10}$$

  RMSprop is a gradient descend variant which is trying to find the global minimum of the error hyperplane.
  Error Function: Mean Absolute Error

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |x_i - x| \tag{11}$$

IV. Results

There are multiple evaluation methods and metrics used during this work. Primarily confusion matrix and metrics derived from it such as ”Accuracy”, ”Sensitivity”, ”Precision”,”Specificity” were used for classification evaluation. For regression evaluation R^2 metric and for calculating error ”Mean Absolute Error” has been used. These metrics were chosen according to the problem.

- **Dataset**
  Our work has been conducted on Ames Housing Dataset [8] which is a free distributed dataset and it can be downloaded online. It has been constructed by Dean De Cock from Truman State University as an alternative to the famous Boston Housing Dataset by Harrison and Rubinfeld (1978). Ames Housing Dataset, consists of a total 79 features which describes the subjected residents in Ames/IOWA. Target feature in this dataset was chosen as Sale price of these samples as we are trying to predict the price trends. Dataset was split into half by 1460 to 1458 samples as training and test.

- **Preprocessing Methods Evaluation**
  After a detailed inspection of data ”Nan” values has been dropped initially but after further steps the definition of ”Nan” values has became more clear as they represent when there is no instance of the feature on the sample. Regarding this information ”Nan” values have been replaced by ”0” and prevented data loss.
  As scaling, Normalization and Standardization results have been compared. For Standardization changes the variance of the data Normalization has been chosen. By Normalization original shape of data has been stored.

Lastly a correlation matrix between features have been constructed. Correlation can have both negative and positive value but desired property would be the absolute value.

According to correlation matrix, a correlation threshold of 0.2 has been chosen regarding the information, 0 means ”features are independent” and 1 means ”maximum dependency of the features”. Using the threshold value (0.2) data-set has been refined to a lesser dimension of 22.

- **Classification Evaluation**

  Before applying classification methods, artificial classes created from target feature.

  First of all Naive Bayes approach has been applied. Naive Bayes has produced respectable accuracy values such as 81% but on a closer inspection it has been revealed that class imbalances were affecting sensitivity and specificity values negatively so much so that it made the model unreliable as it had 50% average sensitivity.

  Second method was KNN. KNN had remarkable accuracy, precision and sensitivity results every measurement were over 80% even with using different ”K” values. For instance; for ”K” equals 1 accuracy was 82%. With ”K” value at 5 best metric values (90% accuracy, 66% precision, 87% specificity) were observed and larger ”K” values were producing worse results. But going into details, imbalances on class samples and nature of the data-set at hand revealed that KNN was also unreliable. Model had the tendency to ignoring classes with less sample size.

  Lastly for classification methods SVM classifier was used. SVM classifier with a second degree polynomial kernel yielded subpar results. As the function degree get higher the produced results were also improving but SVM was also suffering greatly by imbalanced class samples. Although accuracy and specificity values were amazing, sensitivity values were suffering greatly.

  To conclude classification evaluation, even if the SVM and KNN methods were quite successful, due to the nature of given problem and data-set results are not even close to the regression when converted from class labels back to continuous values. This proves that our ”Artificial Class Creation” method was a bad solution to convert the problem from a regression case to classification case.

- **Regression Evaluation**

  Starting with KNN regression, performance metrics on training data was amazing. If whole data was provided R^2 score can reach up to 0.98 but when using trained model to predict test data, results plummeted as low as 0.35s. These results indicated that KNN needed nearly whole data before making a prediction. It was the expected behaviour considering the principles of KNN.

TABLE I
CLASSIFICATION METHODS EVALUATION

Classification Methods	***Hyper-params.***	***Accuracy***	***Sensitivity***	***Precision***
Naive Bayes	-	0.81	0.45	0.54
KNN K = 5	distance = ”Euclidian”	0.84	0.58	0.47
SVC	f = Polynomial d = 5	0.87	0.6	0.6

K = Number of Neighbors, f = Kernel Function, d=Degree of polynomial function

  SVM regression was the second method that has been applied. SVM regression was applied using polynomial kernel with various degree values. Observations showed that SVM regression had scored 0.84 R^2 score on training data but this did not translate to test data while predictions on test data could only achieve 0.51 R^2

  Lastly a dense neural network has been trained for classification. Neural network consisted of five hidden layers. The whole model can be summarized as:

TABLE II
MODEL: ”SEQUENTIAL”

Layer (type)	**Output Shape**	**Num of Params**
dense (Dense)	(None, 256)	5888
dense_1 (Dense)	(None, 128)	32896
dropout (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_1 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2080
dense_4 (Dense)	(None, 16)	528
dense_5 (Dense)	(None, 1)	17

  Total params: 49,665
  Trainable params: 49,665
  Non-trainable params: 0

  As error function Mean Absolute Error(MAE) has been used and Adamgrad was chosen as optimizer. After initializing model several training sessions with different hyperparameters were constructed. With a high learning rate model would overfit with up to 0.95 R^2 score on training data and run worse on test data predictions. Even with higher regularization rates overfit situation could not be averted. Best results on test data were observed with low learning rate and also low regularization values. Epochs were taken as constants as higher epoch values would result in overfit models. After hyperparameters were tuned, best test data results were computed with DNN. Latest model achieved 0.85 R^2 score on training data and 0.86 R^2 score on test data.

  To conclude regression evaluation. It has been proven that House Price data-set was most suitable for regression methods. Although results on first glance could favor classification methods, closer inspection revealed that accuracy could not be used alone for evaluating models and classification results were deceiving. On regression side all test cases and metrics favor that Dense Neural Networks has surpassed SVM on the task of regression. Final result indicates for given data-set, Dense Neural Networks produced robust and accountable results.

Source codes of this work can be found at https://github.com/NickJackolson/HousingPricesML in form of notebooks.

TABLE III
BEST RESULTS FROM ALL METHODS

Regression Methods	***Hyper-params.***	***Error F.***	R^2
KNN	K = 2 distance = ”euclidean”	-	0.38
SVR	f = Polynomial d = 1 c = 1.0	-	0.56
DNN	λ = 0.01 E = 250 c = 0.1	MAE	0.86

f = Kernel Function, λ = Learning Rate, E = Epochs, D = Regularization Coefficient, K = Number of Neighbors

V. Conclusion

With the emerging ML and DL methods, using hidden patterns in big data in order to make future predictions and inferences have become possible. This situation applies to almost all fields including real estate. So we have conducted this work to see if we could pull out some valuable information or accurate predictions. In our work we have used Naive Bayes, KNN classifier, KNN regression, SVM classifier, SVM regression and ANNs with addition of preprocessing methods.

After evaluating regression and classification methods,it has been concluded that for this particular data-set and problem, a dense neural network with five hidden layers yields the best results for regression, among other methods of classification and regression. Highest observed metrics of trained DNN revealed an 86% success rate. For future, different preprocessing methods and different hyper parameter tuning methods can be chosen in order to achieve better results with SVMs. Overall our argument is that proposed ANN method can be used effectively on predicting housing sale prices.

References

[1] Charles A. Calhoun *OFHEO House Price Indexes : HPI Technical Description*. Office of Federal Housing Enterprise Oversight Washington DC 03/1996.

[2] Karl E. Case, Robert J. Shiller *Prices of Single Family Homes Since 1970: New Indexes for Four Cities*. 09/1987.

[3] Office for National Statistics, Land Registry, Registers of Scotland and Land Property Services Northern Ireland *Development of a single Official House Price Index*. 02/2016.

[4] The Dahn Phan. *KNN Model-Based Approach in Classification*. Macquarie University Sydney, Australia 08/2018.

[5] Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang, Rick Siow Mong Goh *KNN Model-Based Approach in Classification*. Institute of High Performance Computing (IHPC) 11/2017.

[6] Guo, Gongde and Wang, Hui and Bell, David and Bi, Yaxin. *KNN Model-Based Approach in Classification*. 08/2004.

[7] Harris Drucker, Chris J.C. Burges, Linda Kaufman,Alex Smola, Vladimir Vapoik. *Support Vector Regression Machines* . Bell Labs and Monmouth University Department of Electronic Engineering

[8] Dean De Cock *Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project*. Journal of Statistics Education Volume 19, Number 3(2011)

[9] Ratner, Bruce *The correlation coefficient: Its values range between +1/1, or do they?*. Journal of Targeting, Measurement and Analysis for Marketing volume 17, 2009/06/01