House Price Prediction Using Machine Learning Techniques

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

***Abstract*—Forecasting provides an insight about future trends. All fields affected by time can benefit from this property. Housing prices are a good example to this. On this work main focus will be the ability to forecast housing prices as accurately as possible. There exists known methosds for forecasting. The most common used techniques include Principle Component Analysis(PCA) [5]. , Linear Discriminant Analysis(LDA) [6]. , Naive Bayes, K-Nearest Neighbors(KNN) [1]. and Support Vector Machine(SVM) [3] [4]. . variants. These techniques help us bringing out hidden patterns and makes it easier for us to interpret our data. Experiments have been conducted with PCA [5], LDA [6], Naive Bayes, KNN [1], SVMs [3] [4] and Dense Neural Networks on House Price data-set [9] in order to better represent the effects of each method on a real world, time-series data with both categorical and numerical values as features. Comparison of the method evaluations will be the primary contribution of this work.**

***Index Terms*—pca, lda, svm, knn, regression, bayes, naivebayes, classification, neural, network, complex, data, data-set, analysis, pattern, recognition, forecasting, prediction, housing, price**

I. Introduction

Forecasting of all kinds has a great value for economics, weather, politics, and so much more fields of study, it is also a popular problem to deal with. Housing Prices sets up an accurate example in this regard. Forecasting housing prices could produce a better understanding in economic, local and social trends. To reach our goal, there are a lot of pattern recognition techniques which can help us analyze data and come up with valuable, refined information and predictions. Before choosing a Pattern Recognition method data should be closely inspected. Pre-processing and interpretation methods should be chosen according to the data-set features and feature relations with problem specific target value. In this instance Ames Housing data-set [9] often cited as Boston Housing data-set has been used. Our data set has total of 80 features with multiple categorical, numerical labeled and continuous data. 80th feature is actually the target which is pursued to successfully predict.

Features can be listed as; Type of dwelling involved in the sale, general zoning classification, Linear feet of street connected to property, Lot size in square feet, Type of road access to property, Type of alley access to property, General shape of property, Flatness of the property, Type of utilities available, Lot configuration, Slope of property, Physical locations within Ames city limits, Proximity to various conditions, Type of dwelling, Style of dwelling, Rates the overall material and finish of the house, Rates the overall condition of the house, Original construction date, Remodel date, Type of roof, Roof material, Exterior covering material of house, Masonry veneer type, Masonry veneer area in square feet, Quality of the material on the exterior, The present condition of the material on the exterior, Type of foundation the house has, height of the basement, The general condition of the basement, Exposure of walkout or garden level walls, Rating of basement finished area, Unfinished square feet of basement area, Total square feet of basement area, Type of heating, Heating quality and condition, Type of air conditioning, Electrical system type, First Floor area in square feet, Second floor area in square feet, Low quality finished area in square feet (all floors), Above grade (ground) living area square feet, Basement full bathroom count, Basement half bathroom count, Full bathroom count above ground, Half bathroom count above ground, Bedrooms above ground, Kitchens above ground, Kitchen quality, Total rooms above ground, Home functionality type, Number of fireplaces, Fireplace quality, Garage type, Year garage was built, Interior finish of the garage, Size of garage in car capacity, Size of garage in square feet, Garage quality, Garage condition, Garage driveway type, Wood deck area in square feet, Open porch area in square feet, Enclosed porch area in square feet, Three season porch area in square feet, Screen porch area in square feet, Pool area in square feet, Pool quality, Fence quality, Miscellaneous feature not covered in other categories(Elevator, Second Garage, Other, Shed, Tennis court), Value of miscellaneous feature in dollars, The month which the house is sold, The year which the house is sold, Type of sale, Condition of sale and the target feature Sale price.

Above mentioned features certainly require some type of feature selection. Another field of difficulty arises when it is considered that some features for example ”overall quality of house” scales from ”0” to ”10” but another feature ”lot size in square feet” varies from 1300 to 215000 which causes logical and scalar imbalances. Yet another difficulty for pattern recognition is the high dimensionality of the data-set.

On this work the performance metrics of various pattern recognition methods will be compared and observations on how well each method performs on this particular data-set will be discussed.

II. Contributions

- Clear demonstration of how pre-processing methods effect given data-set and the summary of decisive factors on these pre-processing methods.
- An attempt to convert a regression problem into a classification problem using a problem specific mathematical formula. Proposed method may be used on other datasets. Yet to be tested on another data-set.
- Direct comparison between evaluation metrics of Regression methods and Classification methods while using the proposed Artificial Class Construction method.
- Evaluation of classification method success while using Artificial Class Construction and a verdict with observation results.
- Proposition of a successful prediction model using DNN with relatively less layers and trainable variables. Observed values indicate a high success rate with a short training period.

III. Pre-processing

While selecting the Pre-processing methods, nature of dataset at hand should always be in consideration. Not all data-sets are compatible with all Pre-processing methods. Inappropriate Pre-processing methods could have no effect or could even have negative effects.

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 need 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. Which has been solved by replacing ”Nan” values with ”0” values.

B. Normalization [7]

Normalization is a commonly used method before the pattern recognition applications. The goal of normalization is to transform features to be on a similar scale. This operation improves the performance and prevents the unbalanced contribution of feature values. It’s simplicity makes it a go to Pre-processing method.

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

X being the feature/features selected to apply normalization, X_{max} is the maximum value for the selected feature and the X_{min} is the minimum value. By finding range $X_{max} - X_{min}$ then dividing the result of $X - X_{min}$, scaled values $X^{'}$ is produced.

C. Standardization [8]

Standardization is also commonly used method before the pattern recognition applications. With Standardization mean of values within features, will always be zero, and the standard deviation will always be one. The graph of standardized values will have exactly the same shape as the graph of raw data, but it may be a different size and have different coordinates. So basically feature space will be centered on origin point. These properties make standardization a meaningful choice of Pre-processing. Rescaling formula is referred as the z-score formula.

$$Z = \frac{x - \mu}{\sigma} \tag{2}$$

Z = new feature values
x = original feature value
μ = mean of the feature samples
σ = standard deviation of the feature samples

D. Principle Component Analysis [5]

Principle component analysis is another popular Preprocessing method with a specific benefit of dimension reduction ability. PCA operation enables visualization of high dimensional data with meaningful 2D or 3D representations. Basically PCA constructs new variables using eigenvectors and eigenvalues. These values are chosen based on how well they can represent the data. If an eigenvalue of a principle component is high than that means it can be used to symbolize the change in data. PCA consists of four main steps.

1) Compute the mean for every dimension of the whole data-set.
2) Compute the covariance matrix of the whole data-set.

$$cov(X_i, X_j) = \frac{1}{n-1}\sum_{k=1}^{n}(X_{ik} - \bar{x_i})(X_{jk} - \bar{x_j}) \tag{3}$$

3) Compute eigenvectors and the corresponding eigenvalues. (A = covariance matrix, I = identity matrix)

$$det(A - \lambda I) = 0 \tag{4}$$

4) Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d × k dimensional matrix W.

Acquired new matrix W, will be the new feature space for the data-set. PCA seems to have an effect of revealing linear distributions and clusters on simple data-sets.

E. Linear Discriminant Analysis [6]

Linear Discriminant Analysis (LDA) is a very common technique for dimensionality reduction problems as a preprocessing step for machine learning and pattern classification applications. LDA operation enables projection of high dimensional data with meaningful 2D or 3D representations. LDA consists of three main steps. The first step is to calculate the separability between different classes (i.e. the distance between the means of different classes), which is called the between-class variance or between-class matrix. The second step is to calculate the distance between the mean and the samples of each class, which is called the within-class variance or within-class matrix. The third step is to construct the lower dimensional space which maximizes the between-class variance and minimizes the withinclass variance.

F. 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.

Correlation Formula:

$$cov(x,y) = \frac{1}{n-1}\sum_{k=1}^{n}(X_{ik} - \bar{x_i})(X_{jk} - \bar{x_j}) \tag{5}$$

$$r_{x,y} = \frac{cov(x,y)}{n-1} \tag{6}$$

- 1 indicates a strong positive relationship between x,y.
- -1 indicates a strong negative relationship between x,y.

G. Artificial Class Construction

When dealing with continuous data, options for pattern recognition methods are being reduced to regression. The subject data-set has continuous data as target values. This situation demands creative workarounds such as Artificial Class Construction as our terms. For this specific data-set 5 different classes were constructed by using standard deviation of target feature to compute class value intervals. Target feature got grouped into 5 classes with equal range. Equation below dishes out the class number.

$$r = (\frac{x}{2\sigma}) \tag{7}$$

r = class number
x = sample
σ = standard deviation

IV. Classification Methods

Classification is initially a method for predicting class values by putting 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.

A. 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.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{8}$$

$$P(c|X) = \prod_{i=1}^{n} P(x_i|c) \tag{9}$$

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.

B. K-Nearest Neighbors [1]

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.

$$d(x,y) = \sum_{i=1}^{n}\sqrt{(x_i - y_i)^2} \tag{10}$$

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)

C. SVM Classifier(SVC) [3]

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.

$$\sum_{i \in SV} y_i \alpha_i K(x_i, x) + b \tag{11}$$

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

V. 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.

A. K-Nearest Neighbors Regression [1]

K-Nearest Neighbors 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.

$$d(x,y) = \sum_{i=1}^{n} \sqrt{(x_i - y_i)^2} \tag{12}$$

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)

B. SVM Regression(SVR) [4]

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 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.

$$Prediction = \sum_{i \in SV} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{13}$$

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

C. 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

$$\begin{aligned} E[g^2]_t &= 0.9E[g^2]_{t-1} + 0.1g_t^2 \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \end{aligned} \tag{14}$$

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

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |x_i - x| \tag{15}$$

VI. Evaluation Methods and Metrics

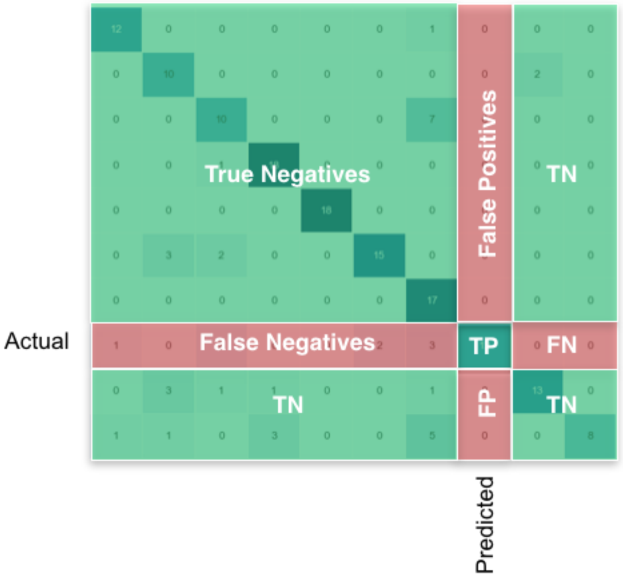
There are multiple evaluation methods and metrics used during this work. Primarily confusion matrix and metrics derived from it such as "Accuracy", "Sensitivity", "Precision","Specificity" were used for classification evaluation. For regression evaluation R^2 metric and for calculating error "Mean Absolute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

A. Confusion Matrix

Confusion Matrix consists of True Positive(TP), True Negative(TN), False Positive(FP) and False Negative(FN) values. Which describe the models behavior on predicting classes. TPs represent predictions of actual positive cases as positive, TNs represent predictions of actual negative cases as negative, FPs represent predictions of actual negative cases as positive and FNs represent predictions of actual negative cases as negative. Using these values, more elaborate metrics such as "Precision", "Sensitivity","Specificity" and "Accuracy" can be calculated. These metrics can be calculated both classwise and modelwise.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

Fig. 1. Multi-Class Confusion Matrix



$$Precision = \frac{TP}{TP + FP} \quad (17)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (18)$$

$$Specificity = \frac{TN}{TN + FP} \quad (19)$$

These metrics enables classwise semantic evaluation of model.

B. Mean Absolute Error

In regression problems there has to be an error function measuring models behaviour in order to optimize the model. For this purpose ”Mean Average Error” function has been chosen. It reflects the absolute distance between sample points and regression function.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - x| \quad (20)$$

C. R^2 Method

R^2 computes the coefficient of determination. It represents the proportion of variance (of target feature) that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse).

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (21)$$

$\hat{y}_i$ = the predicted value of the i^th sample
y_i = the corresponding true value for total n samples

VII. Results

So far each method applied to data-set at hand has been explained briefly. From now on the empiric experimentation results will be discussed. On each step evaluation results were compared and for next step the process with best result has been carried on.

- **Pre-processing 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.
  Dimension reduction methods PCA and LDA has been applied seperately and consecutively but no improvement was observed on any case. Thus PCA and LDA was subtracted from pre-processing. 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.

TABLE I
High Correlated Columns (Correlation > 0.2)

Id	Feature Name	Non-Null Count	Data Type
0	LotFrontage	1460 non-null	numerical
1	LotArea	1460 non-null	numerical
2	OverallQual	1460 non-null	numerical
3	YearBuilt	1460 non-null	numerical
4	YearRemodAdd	1460 non-null	numerical
5	MasVnrArea	1460 non-null	numerical
6	BsmtFinSF1	1460 non-null	numerical
7	BsmtUnfSF	1460 non-null	numerical
8	TotalBsmtSF	1460 non-null	numerical
9	1stFlrSF	1460 non-null	numerical
10	2ndFlrSF	1460 non-null	numerical
11	GrLivArea	1460 non-null	numerical
12	BsmtFullBath	1460 non-null	numerical
13	FullBath	1460 non-null	numerical
14	HalfBath	1460 non-null	numerical
15	TotRmsAbvGrd	1460 non-null	numerical
16	Fireplaces	1460 non-null	numerical
17	GarageYrBlt	1460 non-null	numerical
18	GarageCars	1460 non-null	numerical
19	GarageArea	1460 non-null	numerical
20	WoodDeckSF	1460 non-null	numerical
21	OpenPorchSF	1460 non-null	numerical

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 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.

TABLE II
CLASSIFICATION METHODS EVALUATION

Classification Methods	***Hyper-params.***	***Accuracy***	***Sensitivity***	***Precision***
Naive Bayes	-	0.81	0.45	0.54
KNN K = 1	distance = ”Euclidian”	0.8	0.54	0.53
KNN K = 3	distance = ”Euclidian”	0.82	0.56	0.53
KNN K = 5	distance = ”Euclidian”	0.84	0.58	0.47
KNN K = 7	distance = ”Euclidian”	0.83	0.52	0.44
SVC	f = Polynomial d = 1	0.8	0.49	0.41
SVC	f = Polynomial d = 2	0.85	0.61	0.53
SVC	f = Polynomial d = 3	0.86	0.61	0.55
SVC	f = Polynomial d = 5	0.87	0.6	0.6
SVC	f = Polynomial d = 7	0.86	0.59	0.59

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

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

TABLE III
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

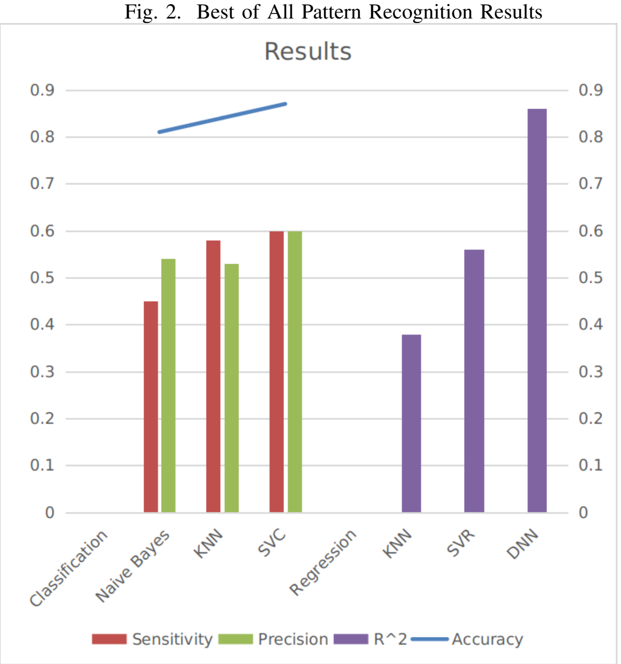
  Total params: 49,665
  Trainable params: 49,665
  Non-trainable params: 0
  As error function Mean Absolute Error(MAE) has been used and Adamgrad was chosen as optimizer. After initializing model several training sessions with different hyperparameters were constructed. With a high learning rate model 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.

TABLE IV
PATTERN RECOGNITION RESULTS

Regression Methods	***Hyper-params.***	***Error F.***	R^2
KNN	K = 1 distance = ”euclidean”	-	0.36
KNN	K = 2 distance = ”euclidean”	-	0.38
KNN	K = 3 distance = ”euclidean”	-	0.36
KNN	K = 5 distance = ”euclidean”	-	0.35
KNN	K = 7 distance = ”euclidean”	-	0.33
SVR	f = Polynomial d = 1 c = 1.0	-	0.56
SVR	f = Polynomial d = 2 c = 0.5	-	0.42
SVR	f = Polynomial d = 3 c = 0.9	-	0.31
SVR	f = Polynomial d = 5 c = 1.0	-	0.34
SVR	f = Polynomial d = 7 c = 1.0	-	0.23
DNN	λ = 0.0001 E = 250 c = 0.2	MAE	0.62
DNN	λ = 0.001 E = 250 c = 0.15	MAE	0.67
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

Fig. 2. Best of All Pattern Recognition Results



VIII. CONCLUSION

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, among other methods of classification and regression. Highest observed metrics of trained DNN revealed an 86% success rate. Source codes of this work can be found at https://github.com/NickJackolson/HousingPricesML in form of notebooks.

ACKNOWLEDGMENT

We thank our Lecturer Ast. Prof. Duygu DEDE ŞENER from Computer Engineering Department on Baskent University (ddede@baskent.edu.tr) for all of her efforts and support.

REFERENCES

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

[2] 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

[3] Sklearn: SVM-Classification, `https://scikit-learn.org/stable/modules/svm.html#classification`

[4] Sklearn: SVM-Regression, `https://scikit-learn.org/stable/modules/svm.html#svr`

[5] Sklearn: PCA, `https://scikit-learn.org/stable/modules/decomposition.html#principal-component-analysis-pca`

[6] Sklearn: LDA, `https://scikit-learn.org/stable/modules/lda_qda.html#linear-and-quadratic-discriminant-analysis`

[7] Sklearn: Normalization, `https://scikit-learn.org/stable/modules/preprocessing.html#standardization-or-mean-removal-and-variance-scaling`

[8] Sklearn: Standardization, `https://scikit-learn.org/stable/modules/preprocessing.html#standardization-or-mean-removal-and-variance-scaling`

[9] 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)