Housing Price Classification

* A CSE802 Course Project

Chang Liu
Computer Science and Engineering
Michigan State University
East Lansing, MI
liucha39@msu.edu

***Abstract*—Housing Price Classification is a subset question of predict the housing price. Sometimes, predicting the price can be quite difficult because of the uncertainty. However, the classification can provide a vogue but much more convincing result.**

***Index Terms*—Classification, Housing price,**

I. Introduction

Housing Price Classification basically separates the prices into different zones: very cheap, cheap, medium, expensive and so on. Here in this classification report, all housing prices are separated into exact 4 classes.

The housing samples here are quite complicated and challenging. First of all, it contains both nominal and ordinal features. As we all known those nominal features are not suitable for most classification techniques such as KNN, SVM, decision tree or perceptron, because we can't calculate the distance among those nominal terms. Second, the features are not conditionally independent in reality and applying Bayesian Classification may perform quite poor because there are so much 'duplicate' information. For example, feature 'State' may includes many 'City' terms; or, 'house style' may be highly correlated with 'age'. Third, there are 1460 samples including the training, validation and test samples. However, we have 4 labels, which means each label has only 365 training samples. So, it's possible to underfit for some relatively complicated model.

II. Dataset Analysis

The dataset of housing price reflects the housing price in Boston, US in certain period. Its author is Dean De Cock from Truman State University. The statistic analysis could be found on $http://jse.amstat.org/v19n3/decock.pdf$ [1]

The dataset contains 79 features and 1460 samples in total. There are 4 kinds of features: location & neighbour, building info, facilities, sale info. Detail information could be found in Table X

The dataset here are separated into training and test set with the ratio of $9:1$. For Bayesian classifiers, since there's no hyper parameter, the experiment cancels the validation set and move it to training set. For other classifiers, cross-validation of 5 fold is applied to find proper parameter for the prediction and accuracy improvement.

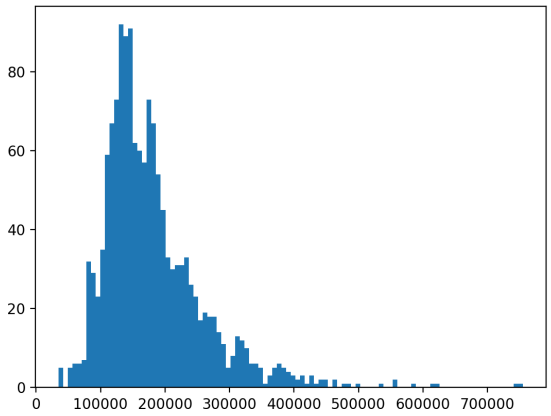


Fig. 1: Raw Price distribution

The price density distribution is as Fig1 shown. We noticed that most of the housing price are in the section of 0 and 400000, although there are some very expensive houses. In the experiment, The prior of each label should not vary too much with each other, so the raw price data are separated into $\{0-135000\}, \{135000-165000\}, \{165000-200000\}, \{200000-\}$, corresponding to cheap, medium, expensive, very expensive.

The total feature can be separated into 43 nominal features and 36 ordinal features. Before the classification, if we check the data, there are 3 nominal features containing large amount of missing values: PoolArea, PoolQC, Fence. It makes sense because maybe most recorded houses in the dataset don't even have a pool, let alone its quality or size. And Large percent of houses don't have the fence. So, before the data cleaning, we remove the above 3 features. And now, we get 40 nominal features and 36 ordinal features.

III. Data Pre-processing/Clean

In order to exam whether the data pre-processing is successful and efficient, the experiment uses the Naive Bayes Classification as a base method. There are 2 reasons for this. First, Naive Bayes has quite simple structure and is not likely to overfit. Another reason is that there is no hyper parameter in Naive Bayes, so we can rule out the uncertainty by unknown parameters.

A. Normalization

There are 4 kinds of normalization used in this report: L1-norm, L2-norm, z-score and Min-Max.

1) L1 and L2 Norm: The L1 is calculated as the sum of the absolute values of the vector. And L2 is calculated as the square root of the sum of the squared vector values. Here the report check both L1 and L2 on different classifiers expect for All kinds of Bayesian classifiers since

$$x' = \frac{x}{||x||_p} \tag{1}$$

$$||x||_p = \sqrt{\sum_{i=1}^{N}(x_i - \overline{x})^p} \tag{2}$$

2) z-score and Min-Max: The z-score normalization is also called standard score, it gives you an idea of how far from the mean a data point is. And Min-Max normalization is the simplest method and rescales the range of features to range in [0, 1]

$$x' = (x - min(x))/(max(x) - min(x)) \tag{3}$$

$$x' = (x - mean(x))/sigma(x) \tag{4}$$

$$\tag{5}$$

B. Dimension Reduction

In the experiment, the data is projected to low dimensional space by PCA and all 4 kinds of Sequential Selection methods: SFS, SBS, SFFS, SBFS.

IV. Classifiers

The base model is Naive Bayesian classifier which could calculate those nominal features. Besides, by modifying the likelihood estimation methods, it also uses Gaussian Bayes model, multinominal Model.

There other classifiers which are tested includes: K-nearest-neighbour, SVM, decision tree and perceptron

V. Experiment of Pre-processing

A. Normalization

To examine whether normalization is good or not, the experiment check it on all classifiers included here.

	Accuracy				
Labels	Level_1	Level_2	Level_3	Level_4	Total
No Norm	0.6268	0.3571	0.5909	0.8571	0.6621
L1-Norm	0.7400	0.4444	0.3580	0.8000	0.5103
L2-Norm	NaN	0.172414	NaN	NaN	0.1724
Min-Max	0.9310	0.2123	0.0000	1.0000	0.3655
Z-score	0.3761	0.0303	0.0000	1.0000	0.2966

TABLE I: Normalization on Gaussian Bayesian classifier

	Accuracy				
Labels	Level_1	Level_2	Level_3	Level_4	Total
No Norm	0.4744	0.2667	NaN	0.7500	0.5517
L1-Norm	NaN	NaN	NaN	0.3103	0.3103
L2-Norm	NaN	NaN	NaN	0.3103	0.3103
Min-Max	1.0000	0.3143	0.1667	0.5422	0.5379
Z-score	0.7636	0.5000	0.6364	0.8974	0.7379

TABLE II: Normalization on SVM classifier

1) Bayes Classifier: The normalization for Bayes classifiers is meaningless literally because we don't need to calculate the distance between features,, all we need is the likelihood, priors and evidence. (posterior can be inferred from above 3 terms).

In TableI, 'NAN' means there's no prediction on current label and '0' means all predictions on the current label are totally wrong.

From TableI, we can conclude that normalization is not useful for the ordinal features in this dataset.

2) SVM Classifier: The SVM classifier is quite sensitive to the unnormalized data, but not all kinds of normalization are suitable for our specific dataset.

In the experiment, TableII show that Z-score normalization performs best on SVM. However, both L1 and L2 normalization perform not well. They tend to predict the housing price to be level4 which means expensive.

B. Dimension Reduction

This experiment apply PCA and all kinds of Sequential Selection methods on different classifiers to check whether dimension reduction is necessary for classification.

Since only applying Dimension Reduction techniques without any normalization can not get a reasonable projection to low dimension space. In the part, Z-score Normalization is applied before those dimension reduction methods.

First we want to show the total accuracy of these dimension reduction techniques, so we use $k = 10$ for PCA and Sequential Selection.

	Accuracy				
Labels	Level_1	Level_2	Level_3	Level_4	Total
Base	0.9310	0.2124	0.0000	1.0000	0.3655
PCA-10	0.7755	0.3571	0.6250	0.7115	0.6552
SFS-10	1.0000	0.4390	0.7143	0.6716	0.6621
SBS-10	1.0000	0.3333	0.7273	0.6286	0.5793
SFFS-10	1.0000	0.3333	0.7000	0.6027	0.5724
SBFS-10	0.9130	0.4130	0.6875	0.7000	0.6414

TABLE III: Gaussian Bayes Dimension Reduction when $k = 10$

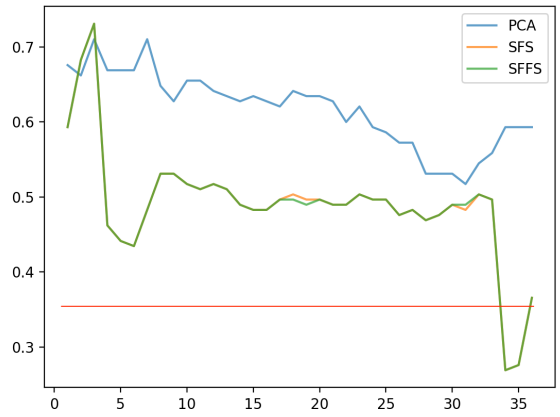


Fig. 2: Gaussian Bayes Accuracy along with $k \in [1, 36]$

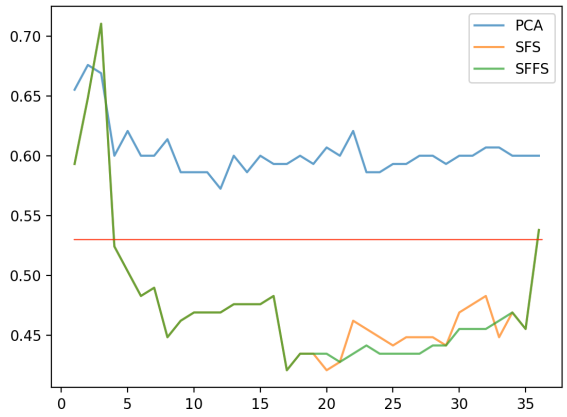


Fig. 3: SVM Accuracy along with $k \in [1, 36]$

	Accuracy				
Labels	Level_1	Level_2	Level_3	Level_4	Total
Base	1.0000	0.3143	0.1667	0.5422	0.5379
PCA-10	0.7143	0.2571	0.4000	0.7255	0.5862
SFS-10	1.0000	0.1579	0.2143	0.4835	0.4897
SBS-10	1.0000	0.1875	0.3333	0.4889	0.4759
SFFS-10	1.0000	0.0800	0.1429	0.4536	0.4345
SBFS-10	1.0000	0.2162	0.0000	0.4944	0.4621

TABLE IV: SVM Dimension Reduction when $k = 10$

	Accuracy				
Labels	Level_1	Level_2	Level_3	Level_4	Total
Naive Bayes All features	0.8043	0.4333	0.6061	0.9167	0.7103
Naive Bayes Ordinal	0.6923	0.4231	0.5667	0.8378	0.6552
Gaussian Bayes	0.7692	0.4783	0.6071	0.8333	0.7103
Multinomial Bayes	0.6042	0.3143	0.4333	0.7188	0.5241

TABLE V: Bayesian Classifiers Test

1) Gaussian Bayes Classifier for dimension reduction: The TableIII shows that dimension reduction techniques can all dramatically improve the accuracy of Gaussian Bayes Classifier. And Figure 3 shows that for Gaussian Bayes Classifier, when $k \geq 8$, the accuracy of all dimension reduction technique decrease, which means that for the ordinal feature of housing price dataset, there are less than 8 features that highly correlated to prices while other features may not be related to price labels. Another thing is for this specific dataset, PCA is better for Gaussian Bayes classifier.

2) SVM classifier for dimension reduction: We check the result of SFS and SFFS after the dimension reduction, when $k \geq 4$, their accuracies both are under the base line $acc = 0.5379$ without any dimension reduction which could be found in TableIV. So for this specific dataset, maybe PCA is better to reduce the feature dimension than Sequential Selection, because if we only choose a few features, it may return us a high bias prediction.

VI. Experiment of Classifiers

In this part, the experiment compares the accuracies of all kinds of classifiers mentioned in the Introduction part including: Bayes, SVM, KNN, decision tree and perceptron.

From the experiment of normalization and dimension reduction, we have learnt the proper techniques to utilize. The following experiments ,except for Naive Bayes Classifier, all utilize the Z-score and PCA (10 components) to do the data pre-processing.

To show the accuracy of labels, the experiment will analysis the confusion matrix. And Cross-validation of 5-folds will be used to examine the variance of the accuracy on different training and validation set. Note Bayesian Classifiers don't utilize cross-validation because there's no hyper-parameter to learn at all.

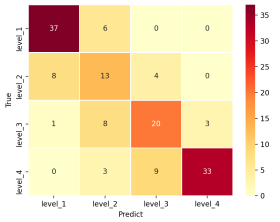
A. Bayesian Classifiers

We can learn from the Table V that Naive Bayesian Classifiers with all features and Gaussian Bayesian Classifiers performs best among Bayesian Classifiers. There re 3 things here. First, the nominal features do help the classifier improve their accuracy. Another thing is that the ordinal features' likelihood density maybe fit the Gaussian distribution better.

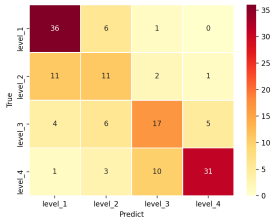
The confusion matrix of all Bayes Classifiers tell us that Level 1 which means cheap housing price is most likely to be predict and Level 4 (expensive) is second. Also, the above 2 labels have relatively high accuracy. However, Level 2 and Level 3 are usually misclassified. One possible reason is that we separate the housing price to keep similar priors but not the same price interval.

B. SVM Classifier

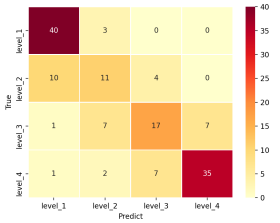
Since we have tested different normalization, for this part, the experiment test SVM Classifiers with different kernels such as: linear, RBF, polynomial and Sigmoid.



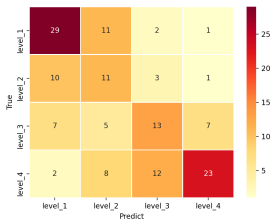
(a) Naive Bayes all features



(b) Naive Bayes ordinal features



(c) Gaussian Bayes

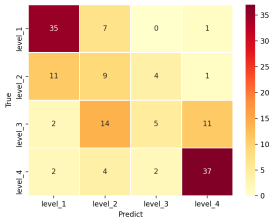


(d) Multinomial Bayes

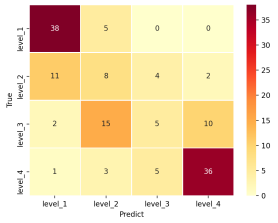
Fig. 4: Confusion Matrix of all Bayesian Classifiers

	Accuracy						
Labels	Level_1	Level_2	Level_3	Level_4	CV-acc	CV-sigma2	Total
RBF	0.7000	0.2647	0.4545	0.7400	0.7009	0.0407	0.5931
Linear	0.7308	0.2581	0.3571	0.7500	0.7063	0.0294	0.6000
polynomial	0.6977	0.3056	0.5652	0.8372	0.6704	0.0494	0.6207
sigmoid	0.6852	0.2963	0.4000	0.6852	0.6104	0.0231	0.5931

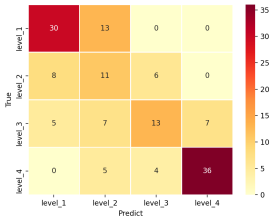
TABLE VI: SVM Classifiers Test



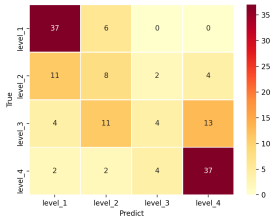
(a) RBF kernel



(b) Linear kernel



(c) Polynomial kernel



(d) Sigmoid kernel

Fig. 5: Confusion Matrix of all SVM Classifiers

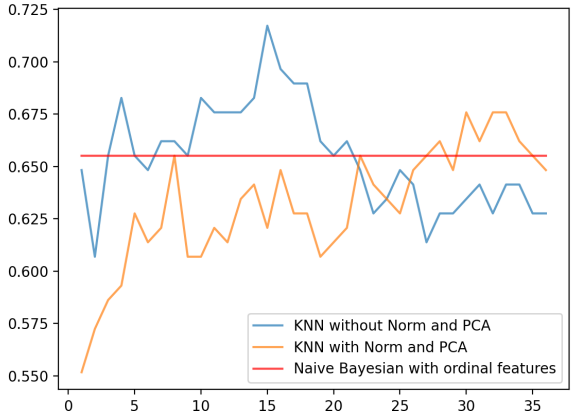


Fig. 6: KNN classifiers with different k

	Accuracy						
Labels	Level_1	Level_2	Level_3	Level_4	CV-acc	CV-sigma2	Total
Decision Tree	0.6857	0.2381	0.3667	0.6842	0.6218	0.0750	0.4897
Perceptron(5,2)	0.5658	0.2000	NaN	0.6406	0.5213	0.1466	0.5862
Perceptron(10,2)	0.5811	0.5000	1.0000	0.6364	0.6255	0.0849	0.6069
Perceptron(20,2)	0.6792	0.2800	0.3333	0.7308	0.7100	0.0333	0.5931

TABLE VII: Multi-layer Perceptron and Decision Tree

The accuracy on Test samples of all SVM kernels are much similar to each other, about 60%, lower than Bayesian Classifiers. But there's one thing interesting that the Linear and sigmoid kernel have much low accuracy variance than RBF and polynomial.

C. KNN Classifiers

The KNN classifiers of ordinal features are lower than twice of the Naive Bayes classifier, but since the accuracy of Bayes is not high, we can't see what it will be when KNN's error rate is imminent towards twice that of Bayesian Classifier.

For this experiment, the KNN without Normalization and PCA performs better than which with when $k \leq 22$. The reason is that some features with large magnitude are also important, so normalization may reduce their weights. But when $k \geq 23$, the KNN classifier with normalization performs better. It means if we have a large k for KNN Classifier, it's better for us to do normalization and feature reduction.

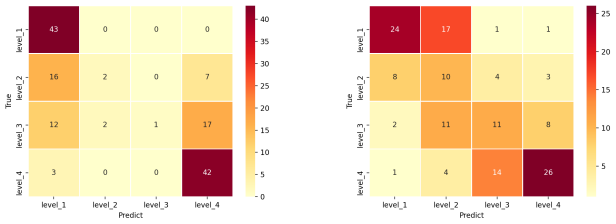
D. Multi-layer Perceptron and Decision Tree

The multi-layer perceptron in this experiment is set to be $(5, 2)$, $(10, 2)$ and $(20, 2)$ size compared to decision tree classifiers.

The decision tree here performs perhaps the worst in all kinds of classifiers with only 49 accuracy. From Figure 7, we can see that the perceptron tends to predict housing price to be either cheap or expensive and very sensitive to data even with normalization and feature reduction.

VII. Ensemble

The experiment uses 3 kinds of Ensemble methods (Bagging, Random Forest, Ada-boost) for Gaussian Bayesian clas-



(a) Perceptron size $(10, 2)$ (b) Decision Tree

Fig. 7: Confusion Matrix of $(10, 2)$ Perceptron and Decision Tree

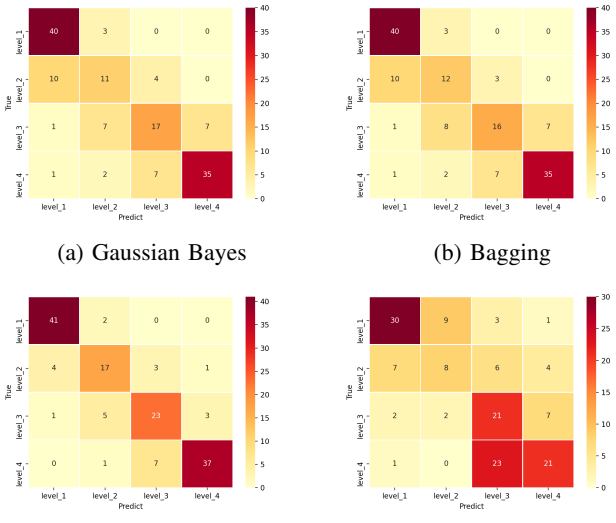
	Accuracy				
Labels	Level_1	Level_2	Level_3	Level_4	Total
Gaussian Bayes	0.7692	0.4783	0.6071	0.8333	0.7103
Bagging	0.7692	0.4800	0.6154	0.8333	0.7103
Random Forest	0.8913	0.6800	0.6970	0.9024	0.8138
Adaboost	0.7500	0.4211	0.3962	0.6364	0.5517

TABLE VIII: Ensemble techniques

sifiers which structure is simple and not likely to overfit the data. The base estimators for these 3 ensemble techniques are set to be 50.

For this specific dataset, bagging achieves nearly the same performance compared with Gaussian Bayesian classifier. However, Adaboost performs not good on both label accuracy and overall accuracy, especially for Level 3 and Level 4.VIII

Random forest achieves the highest overall accuracy among all classifiers with ensembles in this experiment.8. Although the decision tree we test in previous part could only get about 49% accuracy, the random forest could combine different trees to make huge improvement for classification.



(a) Gaussian Bayes (b) Bagging

(c) Random Forest (d) Adaboost

Fig. 8: Ensemble Techniques

VIII. Future Work

First of all, the data set contains much time domain features, but we regard them as ratio magnitude which means these feature may required specific normalization method to make prediction more accurate.

Next, the nominal features are also important but most classifiers can not utilize them. If new methods can be applied to combine those nominal features, we can make better prediction on labels.

IX. Lessons Learned

The ensemble method is very useful. Even we have a bunch of poor classifiers, the ensemble techniques could combine them and return a relatively good result. This is quite helpful when we have a few training samples and suffer from underfitting or overfitting, ensemble could improve performance dramatically.

Another thing is the **No Free Lunch Theorem**. Since the housing prices are separated based on similar priors, some target function could be similar. No matter we want to choose a 'good' classifier or a 'bad' one, the 'good' classifier (Bagging of Gaussian Bayes) may not be superior to 'bad' classifier.

X. Summary

The work in the report can be concluded as Table IX. The Naive Bayesian classifier and test algorithm is self-implemented. Other Classifiers are from python package sklearn [2]

Data Clean	Normalization	Z-score, Min-Max, L1, L2
	Dimension Reduction	PCA, SFS,SBS, SFFS, SBFS
Classifiers	Bayes	Naive Bayes Nominal and Ordinal
		Naive Bayes Ordinal, Gaussian Bayes, Multinominal Bayes
	SVM	RBF, Linear, Polynomial, Sigmoid
	Decision Tree	
	Perceptron	Different Size
	KNN	Different k value with cleaned data
		Different k value with raw data
Ensemble	Bagging	Based on Gaussian Bayes
	Adaboost	
	Random Forest	Based on Decision Tree

TABLE IX: Experiment Content

XI. Appendix

	Features
location&neighbours info	MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour Utilities LotConfig LandSlope Neighborhood Condition1 Condition2
building info	BldgType HouseStyle OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType MasVnrArea ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2 BsmtFinSF2 BsmtUnfSF TotalBsmtSF
facilities	Heating HeatingQC CentralAir Electrical 1stFlrSF 2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual TotRmsAbvGrd Functional Fireplaces FireplaceQu GarageType GarageYrBlt GarageFinish GarageCars GarageArea GarageQual GarageCond PavedDrive WoodDeckSF OpenPorchSF EnclosedPorch 3SsnPorch ScreenPorch PoolArea PoolQC Fence MiscFeature MiscVal
sale info	MoSold YrSold SaleType SaleCondition

TABLE X: Feature Types

References

[1] D. D. Cock, "Ames, iowa: Alternative to the boston housing data as an end of semester regression project." Truman State University, 2011. [Online]. Available: https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," vol. 12, 2011, pp. 2825–2830.