Supervised Learning Model Comparison

Zhilan Deng

***Abstract*-** This paper will perform five different supervised learning algorithms on two classification problems: 1) Speaker Accent Recognition and 2) determine whether a person makes over 50K a year. The five supervised learning algorithms are:

1) Decision tree
2) Neural network
3) Boosting
4) Support Vector Machine (SVM)
5) K-nearest Neighbors (KNN)

I. Introduction

Classification problems

This paper selects two classification problems: 1) the recognition of a speaker's accent and 2) prediction of a person's income based on his/her census statistics.

This paper selects these two classification problems because human could easily make mistakes in trying to solve these two problems. Accent is relatively a subject point of view in most of the time, however, by extracting quantitative statistics using Mel-Frequency Cepstral Coefficients (MFCCs) on the original time domain soundtrack of the maximum 1s of reading of a word[1], the accent from different area could be easily recognized. And in predicting the income of a person, different people have different opinions on which variable would be the most important factor in impacting the income of a person and by applying supervised learning algorithms on this problem, the answer to this question may be able to answered, and shows which factor or factors would be the most important one in earning higher income.

The other reason this paper selects these two classification problems is because of the structure of the two datasets. Accent dataset have only 330 data points and 12 variables, all 12 variables are quantitative, and the response variable is multiclass. On the contrary, the income dataset has 32562 data points and 12 variables, and the response variable is binary. Different from the quantitative only variables in accent dataset, among the 12 variables in the income dataset, 6 of them are category variables which are neither in number nor with some order sequence, the detail will be discussed later in the following section. By using these two very different datasets, the comparison between different supervised learning algorithms would be more obvious.

Accent Dataset

Accent dataset is generated using Mel-Frequency Cepstral Coefficients (MFCCs) performing signal feature extractions on the recordings of one English word of the speakers. The response variable is the accent: US, ES, FR, GE, IT and UK. the factors are the 12 statistics produced by MFCCs. There are 165 data points classified as US, 30 data points as Spain, 30 data points as France, 30 data as from Georgia, 30 data points as Italy, and 45 data points as UK.

An example of the data points from the accent dataset is shown in table 1.

Table 1. the data points from accent dataset.

language	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
ES	7.07	-6.51	7.65	11.15	-7.66	12.48	-11.71	3.43	1.46	-2.81	0.87	-5.24
ES	11	-5.16	3.95	11.53	-7.64	12.14	-12.04	3.49	0.6	-4.51	2.33	-6.22
FR	-2.3	0.965	2.79	1.2	-9.31	7.93	-10.13	6.77	-3.7	-2.82	1.68	-4.83
FR	2.29	-4.91	-1.13	2.714	-11	6.362	-9.913	4.69	-2.8	-2.03	6.08	-4.81
GE	-2.9	2.407	-0.82	3.177	-4.95	10.11	-12.87	6.81	-1.6	-1.8	4.22	-4.78
GE	4.24	-0.09	-1.36	4.95	-6.54	9.744	-12.29	4.82	-0.1	-1.97	4.13	-7.77
IT	-1.9	-3.48	5.14	7.246	-7.2	8.623	-11.38	3.75	0.38	-1.66	0.6	-2.2
IT	2.69	-5.05	3.02	6.262	-10	8.518	-11.36	1.47	1.31	-3.64	-0.8	0.463
UK	1.54	-7.16	3.78	6.719	-9.35	8.091	-11.78	3.12	-0.4	-4.25	-0.9	-5.71
UK	3.98	-3	0.03	9.453	-8.13	10.15	-9.987	1.36	-0.4	-3.75	1.38	-4.73
US	4.8	-0.33	-1.99	8.256	-7.62	8.342	-7.3	3.41	-1.2	-0.88	3.77	-4.28
US	0.16	-1.33	2.45	7.387	-4.63	10.41	-11.7	3.5	0.1	-1.99	0.99	-4.63

Income Dataset

The income dataset is much more straight forward than the accent dataset, it includes the census attributes and the income of a person. The response variables is whether a person's income is larger than 50, 000, and the factors are the different census attributes of that person: age, work class, education, marital status, occupation, relationship in the family, race, sex and working hours per week. One thing to be noted here is that the data is extracted in 1994, so it will only represent the facts in 1994.

Some examples of the data points from the income dataset are shown in table 2.

Table 2. the data points from income dataset

age	work_class	fngwgt	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_perweek	income
39	State-gov	77516	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	0
50	Self-emp-not-inc	83311	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	0
38	Private	215646	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	0
53	Private	234721	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	0
28	Private	338409	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	0
37	Private	284582	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	0
49	Private	160187	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	0
52	Self-emp-not-inc	209642	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	1
31	Private	45781	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	1

II. SUPERVISED LEARNING ALGORITHMS

OVERVIEW

All the supervised learning processes are in R environment. One thing to be noted here in the income dataset is that there are 75.92% of the data points have income less than 50k, and the algorithms can easily classified all the data points to less than 50K and reach the accuracy to 75.92%. To avoid this result causing by the biased dataset, this paper randomly selects 10000 data points among the data points where the income is less than 50k to make 56.66% of the data points classified as income less than 50k and 43.34% as income more than 50k. the data ranges also varies much in the income dataset, and this paper also scales the also qualitative variables into (0, 1) to get a better performance.

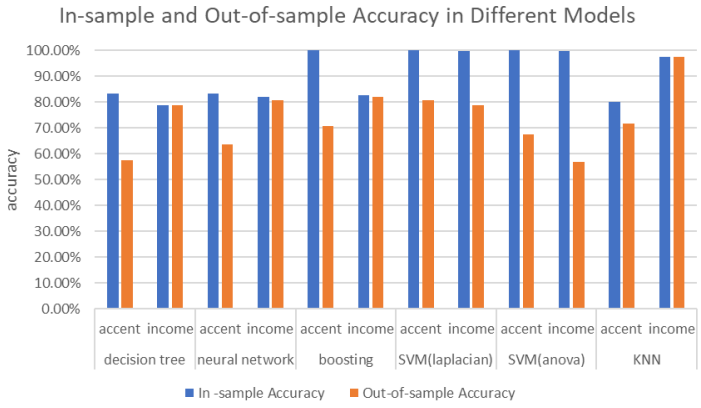
Finally, for each algorithm, the data split the whole dataset into a training dataset and a test dataset. Impact of different training set size will be discussed later in this paper.

OVERALL ACCURACIES IN DIFFERENT MODELS

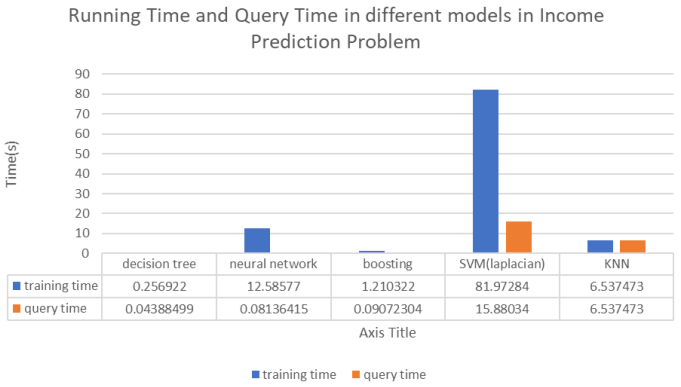
The in-sample and out-of-sample accuracies and running time in different models applying to the two classification problems are shown in figure 1.

Some algorithms perform better among other algorithms such as KNN for income prediction and SVM for accent recognition.

There are also some overfitting exists in the models which can be showed from the differences between the in-sample and out-of-sample accuracies. The overfitting is more likely to exist in accent recognition problem rather than the income prediction problem, especially in decision tree, neural network and boosting algorithm where the differences between the in-sample and out-of-sample accuracies are larger than 20% in accent recognition problem but smaller than 0.2% in the income prediction problem.



(a)



(b)

	decision tree		neural network		boosting		SVM(laplacian)		SVM(anova)		KNN	
	accent	income	accent	income	accent	income	accent	income	accent	income	accent	income
In -sample Accuracy	83.48%	78.85%	83.48%	82.10%	100.00%	82.71%	100.00%	99.72%	100.00%	99.72%	80.00%	97.65%
Out-of-sample Accuracy	57.58%	78.77%	63.64%	80.85%	70.71%	82.02%	80.81%	78.81%	67.68%	56.78%	71.72%	97.56%

(c)

Figure 1. The accuracy and running time of different models in two classification problems, (a) and (c) is the graph and the table showing the accuracies, (b) is showing the training time and query time in different models in income prediction problem.

The main reason of the overfitting in accent database is that there are not enough data points for this problem. The total number of data points is only 330, and when split the dataset into training and testing sets, the number of data points in the training set decreased again which makes even less data points into the training process.

One way to solve this problem is by introducing cross validation where we split the training dataset into K-fold and run training and validation in turns on some (k-1)folds and 1 folds separately and repeatedly to get a less overfit model with limited data points.

In terms of the running time, this paper only uses the income dataset to show the running time since it has more data points and can show the time differences between models more obviously. One thing to be noted is that, for KNN algorithm, the algorithm used in this paper does not separate training time and query time, so the time in KNN column is the total time of training and query. From the figure 1(b), decision tree has the shortest training time and query time while SVM has the longest training and query time, especially for income prediction problem, the training time is over 1 minutes which is much longer than any other algorithms.

CROSS VALIDATION

Because of the existence of overfitting and limited data points in accent dataset, this paper introduced cross validation in the algorithms to solve the overfitting problem.

The performance of cross validation in decision tree, neural network and boosting algorithm is shown in figure 2. Generally, the accuracy of testing dataset is increasing and the distance between the in-sample accuracy and out-of-sample accuracy is shortened while the number of folds increasing, which can be seen clearly when k increased from 3 to 5 in neural network.

Introducing cross-validation improved the overfitting problem in some extent in decision tree model and neural network model. But considering the limited improvement and the increasing running time, this paper does not implement cross validation in the models.

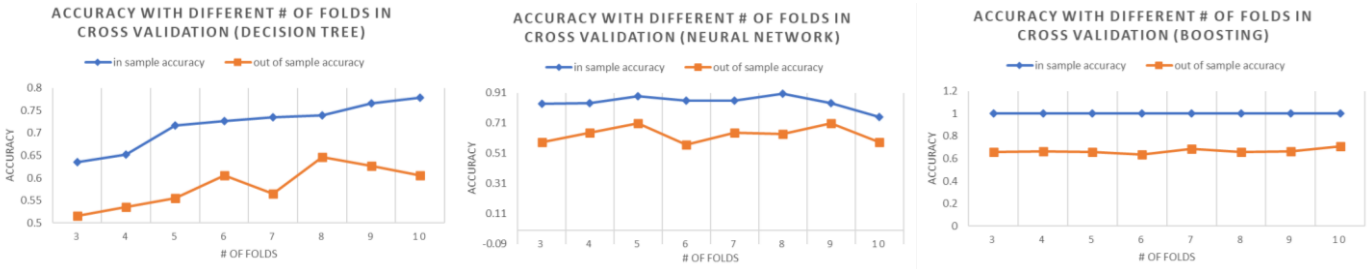
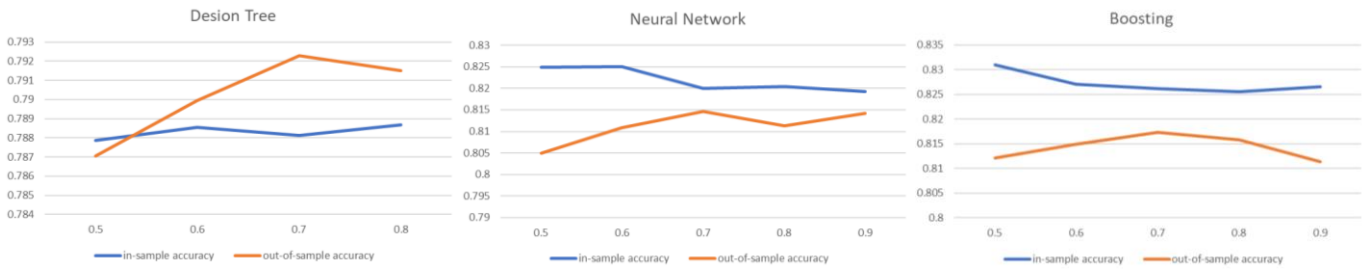


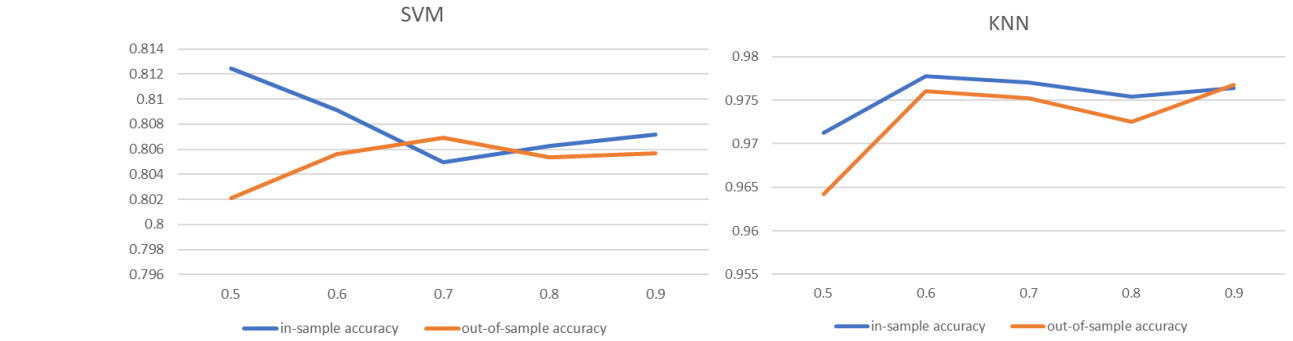
Figure 2. the in-sample and out-of-sample accuracy in accent dataset after introducing cross validation with different # of folds.

TRAINING SET SIZE

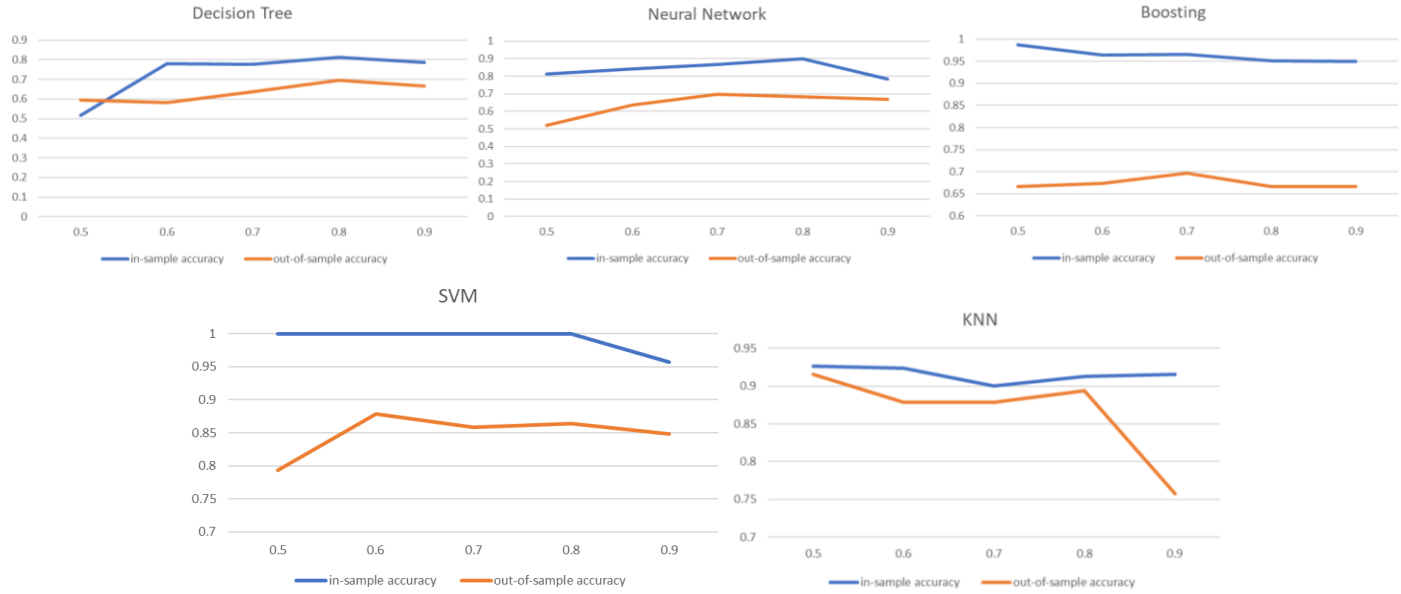
The accuracies under different training sets are shown in figure 3.

When training set size is small, for example 50% of the whole dataset, it is most likely an overfit will occur because of the limited data points, and the distance between in-sample accuracy and out-of-sample accuracy will be large than with the increase of the training set, the distance between the in-sample and out-of-sample accuracies will decrease. This pattern can be observed in all the graphs in figure 3 except the decision tree algorithm in both problems and KNN algorithm in accent recognition problem. For the decision trees in both algorithm, both accuracies are low when the training set size is small and that might be the reason why the overfit is not showing since the model performs bad both in sample and out of sample. For KNN algorithm in accent recognition problem, the overfit occurs when the training set size is large, for example 90% of the whole dataset, this overfit however, might be a false negative sign, since in this situation the test set has only 33 data points, so even only 1 data point was classified in the different class, the error rate is 3%.





(a)



(b)

Figure 3. accuracies under different training set size from using 50% of the whole dataset to 90% of the whole dataset: (a) the income prediction problem, (b) the accent recognition problem.

RESULT ANALYSIS & HYPERPARAMETER TUNING

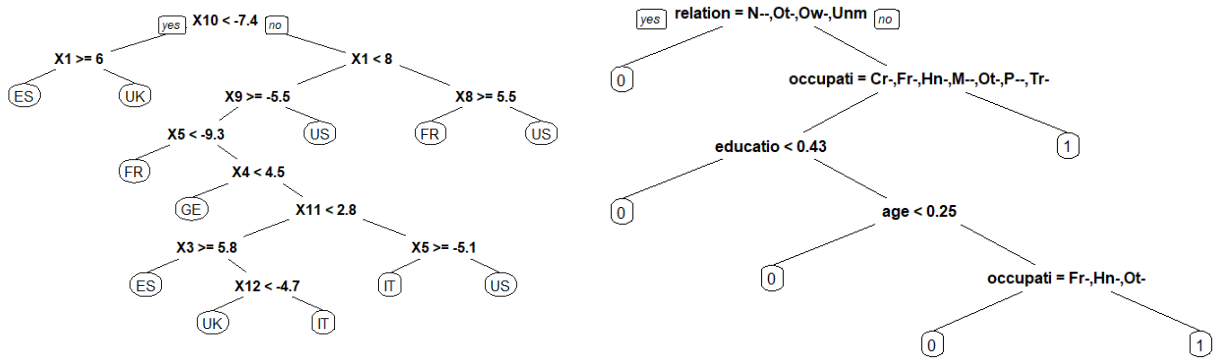
DECISION TREE

The original decision tree without tuning is shown in figure 3.

The accent decision tree is hard to interpret since the variables x1, x2, …, x12 are just different signal extracted using MFCC.

The decision tree of the income prediction, however, is much easier to interpret than that of accent recognition. We could easily see which group would earn more income among others from the decision tree. Interestingly, the relationship in the family is the root node of this decision tree, which means that it has a heavy impact on income and it shows that if a person not in a family, or own child, or unmarried, or in any other relationship other than husband and wife, the person would have little possibility to earn income more than 50K in 1994. Not surprisingly, people with higher education level, older age tend to earn more income. And the split on the occupation also correspond to the average salary of different occupations[2]. The occupations that the decision tree classified as income less than 50k are circled with a red box in the graph in figure 2, it is not hard to observe that those occupation groups are among the occupation groups with lowest average salaries.

.



(a) (b)

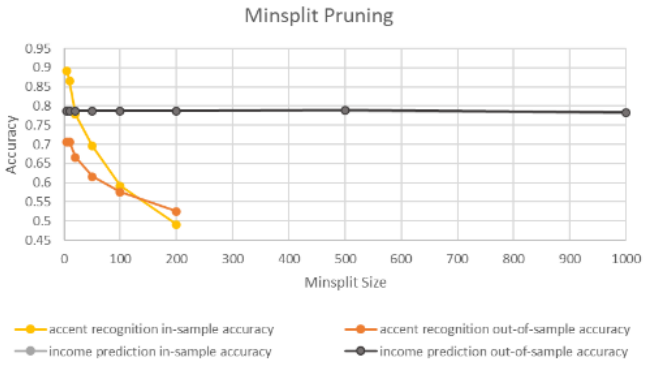
Figure 3. the decision tree of accent recognition (a) and income prediction (b). In graph b, N- represents Not in family, Ot- Other relative, Ow- - Own child, Unm – Un-married, Cr- - Craft repair, Fr- - Farming & Fishing, Hn- - Handless & Cleaner, M- - Machine operation inspect, Ot- - Other Service, P- - Private house service, Tr- - Transport moving.

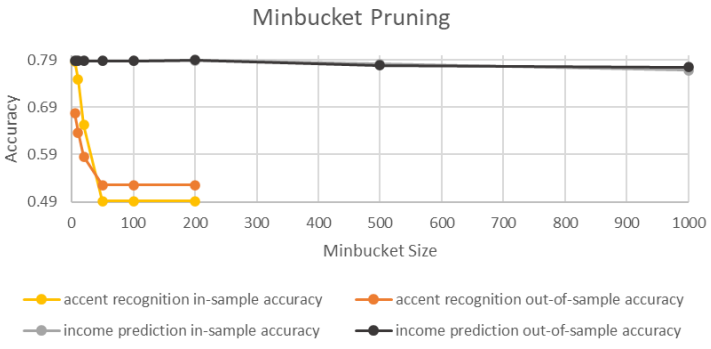
In decision tree, the hyperparameters this paper tuned are the prune methods, this paper apply prune trees based on minsplit, minbucket and cp values. The result of the accuracies under each prune method is shown in figure 4.

Minsplit and minbucket are similar, minsplit is the minimum number of observations that must exist in a node in order for a split to be attempted [3], while minbucket is the minimum number of observations in any terminal node, these two are the pre-pruned methods in Decision Tree. The tree will not split a branch of a node if the number of the observations are less than minsplit or the number of the terminal node after split is less than minbucket value. This paper compared different minsplit size and minbucket size from 10 to 1000 by looking at the in-sample and out-of-sample accuracies in two classification problems.

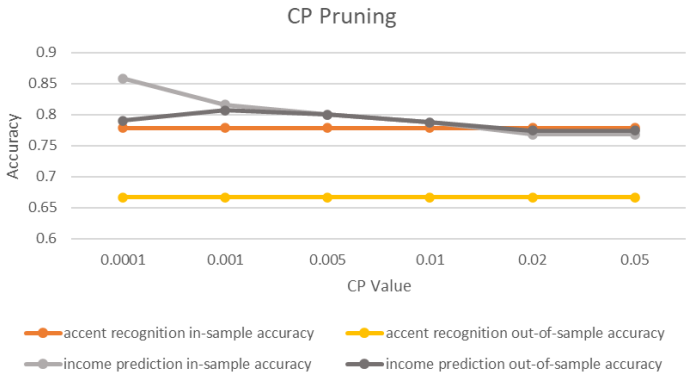
From graph4(a) and graph4(b), the impacts of minsplit and minbucket on income prediction problem are not obvious while their impacts are obvious in accent recognition problem because of its limited data points. Also shown in figure 4, as the sizes of the minsplit and minbucket increase, the overfitting decreases in accent dataset because larger numbers of minsplit and minbucket avoid small trees specifically fit a small number of nodes only, thus avoid the overfit to only some specific cases in the dataset.

On the contrary, the more sensitive dataset to the different cp value is income dataset. Cp value is the complexity parameter value, it decides if the split will be attempted, any split that does not decrease the overall lack of fit by a factor of cp is not attempted. Figure 4(c) shows that similar to minsplit and minbucket, lower cp value introduces overfitting in the model, when the cp value increases, the overfitting decreases in the income prediction problem.





(a) (b)



(c)

Figure 4. the accuracies in two classification problems in different pruning method: (a) different minsplit sizes, (b) different min bucket sizes, (c) different cp values.

NEURAL NETWORK

In neural network section, this paper used nnet library in R. The hyperparameters this paper tuned on are the number of hidden units and the stop criteria, the results of the impact of these two hyperparameters are shown in figure 5.

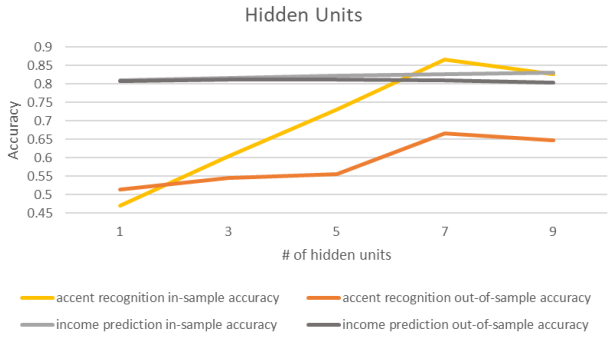
Increasing the hidden units increases the accuracy of both in-sample and out-of-sample in accent dataset but decrease the accuracy in out-of-sample in income dataset (see graph(a) in figure 5). And it is also obvious that increasing the number of hidden units will increase the overfit in the model, so it is important to find the appropriate number of hidden units in neural network while not the larger number of hidden units.

In terms of stopping criteria, there are three stopping criteria in this nnet function: (1) maximum number of iteration, (2) fit criterion falls below a certain value, and (3) optimizer is unable to reduce the fit criterion by a factor of at least 1 – some certain value[4].

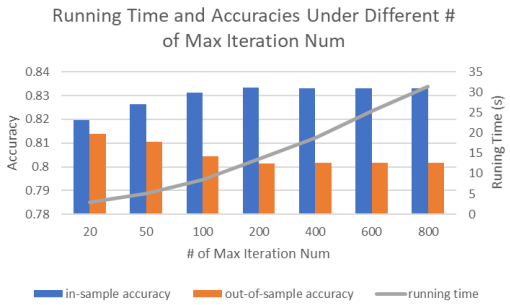
The advantage of using maximum number of iteration as the criteria is that the method will be simple with no request of converge of the final model, but the disadvantage is that the number of maximum iteration times is hard to decide. Smaller number of max iteration times might lead to a unmatured model without enough training, but the large number of max iteration times might cause some redundant iterations and longer running times, especially when the dataset is large. Figure 5(b) shows how the running time increases when the maximum number of iteration times changes from 20 to 1000 in the income dataset. One Thing to be noted is that, after iterating 200 times, the accuracies stay the same which means that the model has already converged, any more iteration is useless in this model. If only set the maximum number of iterations as the stopping criteria, the number need to be decided carefully to avoid an unmatured model while not having too many redundant iterations in training. Also to be noted that the larger number of max iteration num introduced overfitting in the model.

This leads to the second and third stopping criteria relate to the convergence of the model. If selecting these two criteria as the stopping criteria, we could make sure that there will be no more redundant iterations in the model and the output model would most likely to be the perfect model if converged. However, these two stopping criteria depends on the values the user put in and the attributes of the original dataset, if the number is too small or the dataset cannot converge in neural network model, the model may fall in an infinity iteration situation. But if the number too large, then the model will stopped training before it leans enough.

So in the neural network model applied in this paper, all three stopping criteria are set, and the model will stop when meets the maximum number of iterations if the model doesn't converge in earlier steps.



(a)



(b)

Figure 5. The hyperparameter tunning in neural network. (a) changing the number of hidden units, (b) changing the number of max iteration Num

	var <fctr>	rel.inf <dbl>
relationship	relationship	31.0314617
occupation	occupation	16.0953751
marital_status	marital_status	14.7095933
education_num	education_num	13.7781096
age	age	10.6829759
hours_perweek	hours_perweek	5.3016204
fngwgt	fngwgt	3.2798632
race	race	2.5052662
work_class	work_class	1.7443343
sex	sex	0.8714002

(a)

	var <fctr>	rel.inf <dbl>
X10	X10	19.095883
X1	X1	15.972718
X7	X7	10.479432
X2	X2	8.089006
X9	X9	7.097780
X4	X4	6.999246
X5	X5	6.403886
X8	X8	6.054094
X3	X3	6.012419
X6	X6	5.567084

(b)

Figure 6. the result of applying boosting algorithm on income prediction (a) and accent recognition (b).

BOOSTING

The results of applying boosting algorithm on the two classification problems are shown in figure 6.

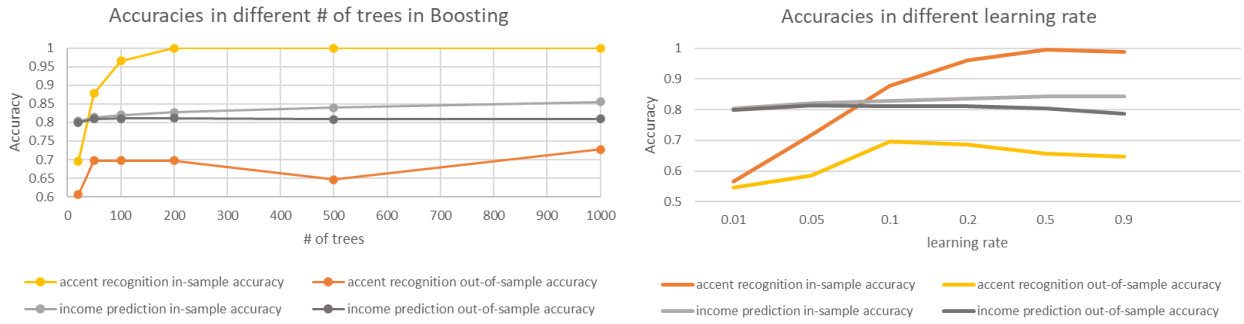
Similar to the results in Decision Tree algorithm, for the income prediction problem, the most important variable is the relationship in the family and the important variables also include occupation, marital status, educational level and age, which are the variables used in the tree in figure 3(b). For the accent recognition problem, x10 is the most important variable and is also the root node in the tree built by decision tree in figure 3(a). Different from the variables in income dataset, the relationship values between the input variables and the response variable in accent recognition problem do not vary much, and every input variable has some relationship to the response variable to some extent.

The Hyperparameters this paper tuned in boosting algorithm are the number of tree used and the learning rate, the result of the tunning are shown in figure 7.

For the number of trees used in this model, among all the other hyperparameter tunings in this project, this hyperparameter is the one to introduce the overfit most quickly. In figure 7(a), when the number of trees used increases from 1 to 200, the in-sample accuracy of the accent recognition problem increased very quickly from 0.69 to 1 and stay 1 when the number of trees used increases. But the out-of-sample accuracy stay in the same level around 0.7 in the whole process. The pattern in the income prediction problem also shows that when number of trees used in the model increases, the in-sample accuracy increases faster than the out-of-sample accuracy and lead to the overfit in the model.

When the learning rate is small, the tree learns very slow, and in small dataset, such as the accent dataset, the tree will probably stop learning before it learns enough. This shows in figure 7(b) where the accuracies are low in accent recognition problem when the learning rate is small. When the learning rate is too large, the model will be unstable and cannot preserve the previous model after training which will also lead to the decreasing of accuracy and is shown in figure 7(b) where the accuracies decreases when learning rate is larger than 0.2.

For the two hyperparameters, both # of trees and learning rate value need to be set appropriately since too small or too large values in these two hyperparameters may cause lower accuracies and higher overfit in the model.



(a) (b)

Figure 7. the accuracies in tuning boosting: (a) different # of trees, (b) different learning rates

SUPPORT VECTOR MACHINE(SVM)

The coefficients generated from support vector machine are shown in figure 8. In the SVM model applied to accent recognition problem, the variable which has the most impact on the response variable is x10, and that is the same from the boosting algorithm. As for the income prediction problem, SVM model shows a different result compared to decision tree or boosting. In SVM algorithm, the variable has the most impact on the income is the race, and it shows that both white race and black race have a negative impact in earning high income. In terms of marital status, married people still have positive impact in earing high income while un-married people has an negative impact on income earning. And for the factor of sex, male seems to have better possibility to earn high income compared to female from what the model generated from the dataset. And lastly the impact of occupation also matches with the average salaries of major occupation groups in [2].

In SVM algorithm, this paper compares different kernels used in the model: gaussian kernel (rbfdot), polynomial kernel (polydot), Laplacian kernel (laplacedot), and Anova RBF kernel (anovadot). The accuracies of different kernels are shown in figure 9.

Different from all the other algorithms, in all four kernels used in this paper, the performances on accent recognition problem are better than the performances on income prediction problem. There are two main reasons of this performance: 1) SVM does not perform very well on large noisy dataset. It will be hard for SVM to separate the large number of data points and select a function, and 2) SVM does a better job on quantitative variables compared to qualitative variables since the separation line or hyperplane put by the SVM model is based on the Euclidean distances.

And as we have showed in previous section, the running time of SVM on large dataset is substantially longer than other algorithms. So from what showed in the experiments in this paper, SVM is not suitable for large noisy database with a great number of qualitative variables, which is the income dataset in this paper.

x10	x7	x9	x8	x11	x5	x6	x2	x3	x4
23.3363433	17.4560602	7.1690473	2.5608670	0.5079142	-0.2525424	-2.0731705	-2.2242607	-4.2168029	-9.0134473

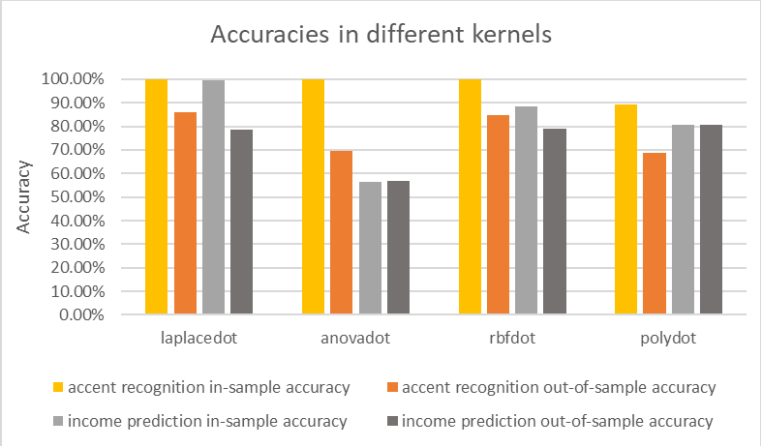
x1	x12
-9.1799693	-13.3327718

(a)

marital_status.Married.AF.spouse	marital_status.Married.civ.spouse	relationship.Wife
1.52628013	1.52411140	1.02156665
work_class.Federal.gov	sex.Male	education_num
0.65129037	0.62467126	0.47721263
work_class.Self.emp.inc	work_class.Private	age
0.46268719	0.31591182	0.31095798
hours_perweek	occupation.Protective.serv	occupation.Exec.managerial
0.29759902	0.24630531	0.20878695
race.Asian.Pac.Islander	occupation.Tech.support	relationship.Not.in.family
0.18711156	0.14980727	0.13801079
relationship.Unmarried	occupation.Prof.specialty	work_class.Local.gov
0.10509877	0.08812089	0.08515640
occupation.Sales	marital_status.Widowed	fnlwgt
0.07847816	0.05816545	0.04812984
work_class.State.gov	relationship.Other.relative	work_class.Self.emp.not.inc
0.01267349	0.00000000	-0.01785574
marital_status.Married.spouse.absent	relationship.Own.child	marital_status.Never.married
-0.02673362	-0.15360490	-0.18224609
marital_status.Separated	occupation.Craft.repair	occupation.Transport.moving
-0.26234693	-0.26510498	-0.50301349
occupation.Armed.Forces	occupation.Machine.op.inspct	occupation.Handlers.cleaners
-0.80392828	-0.88650763	-1.17700413
occupation.Other.service	occupation.Farming.fishing	work_class.Without.pay
-1.31788663	-1.40991405	-1.50986354
occupation.Priv.house.serv	race.White	race.Black
-1.80553681	-3.51083579	-3.57287393
race.Other		
-3.89006680		

(b)

Figure 8. the coefficient generated from SVM: (a) accent recognition problem using laplacedot, (b) income prediction problem using polydot.



(a)

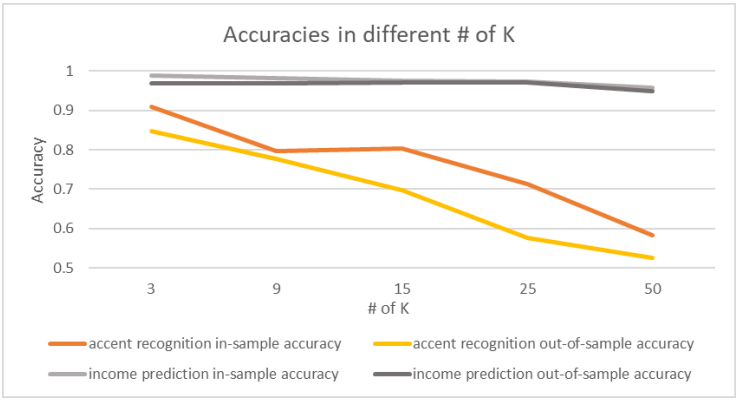
	accent recognition		income prediction	
kernel	in-sample accuracy	out-of-sample accuracy	in-sample accuracy	out-of-sample accuracy
laplacedot	100.00%	85.86%	99.72%	78.56%
anovadot	100.00%	69.70%	56.62%	56.78%
rbfdot	100.00%	84.85%	88.42%	79.05%
polydot	89.13%	68.69%	80.51%	80.75%

Figure 9. the in-sample and out-of-sample accuracies in different kernels

K-NEAREST NEIGHBOR(KNN)

KNN is relatively a simple model but efficient algorithm to apply, it looks at the data points k-nearest neighbors and return the prediction based on the value of that k-nearest neighbors.

The hyperparameter tuned in this paper for KNN algorithm is the number of K, and the results are shown in figure 10. In accent recognition problem, when k increases, both in-sample accuracy and out-of-sample accuracy decreases. In income prediction problem, when k increases, the in-sample accuracy decreases but the out-of-sample accuracy increases which shows that the overfitting is decreasing while k increases. The different performance in these two datasets shows that the most appropriate number of K varies for different dataset and different problems, if the dataset is small, the a small number of k might be a better choice, but when the dataset is large, the bigger number of K might perform better and have less overfit in the model.



(a)

	accent recognition		income prediction	
k	in-sample accuracy	out-of-sample accuracy	in-sample accuracy	out-of-sample accuracy
3	90.87%	84.85%	98.84%	96.94%
9	79.57%	77.78%	98.25%	96.90%
15	80.43%	69.70%	97.48%	97.05%
25	71.30%	57.58%	97.30%	97.20%
50	58.26%	52.53%	95.68%	94.81%

(b)

Figure 9. accuracies in applying different K values in KNN.

CONCLUSION

This paper applied different supervised learning algorithms on two very different datasets and problems, one dataset has limited data points with multiclass response variables, and the input variables are all quantitative, the other dataset has large number of data points with a binary response variable, and the input variables are a mix of quantitative and qualitative variables.

In the experiments above, decision tree, neural network, boosting and KNN algorithm perform better on large dataset while SVM performs better on smaller dataset. Tuning the hyperparameters is also very important to get better performance and tuning the hyperparameter is finding the balance between a higher performance and a lower overfitting in the model.

In terms of the best model for these two problems. For accent recognition problem, SVM is the best model in terms of the performance, it has the highest performance and the lowest overfitting, and the running time is also affordable for 330 datapoints. For income recognition problem, KNN is the best model. Figure 1(a) shows that KNN has the highest accuracy in both in-sample and out-of-sample datasets and the overfit is also low in KNN, considering only 6 seconds running time for over 30000 data points, KNN would be the best algorithm for income prediction.

REFERENCES

[1] Fokoue, E. (2020). UCI Machine Learning Repository [[Web Link]]. Irvine, CA: University of California, School of Information and Computer Science. W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[2] https://www.statista.com/statistics/218235/median-annual-wage-in-the-us-by-major-occupational-groups/

[3] https://www.rdocumentation.org/packages/rpart/versions/4.1-15/topics/rpart.control

[4] https://www.rdocumentation.org/packages/nnet/versions/7.3-14/topics/nnet