ASSIGNMENT -2 SOLUTION

CS422 -Data mining

Harsha and Rutvik TA solution

1

1.1

Dataset	Total Attributes	Attribute Types
Iris	5	Numeric-4, Nominal-1
Vote	17	Nominal-17
Diabetes	9	Numeric-8, Nominal-1

IRIS : dataset consists of 150 Iris flowers, 50 each FROM : Setosa, Versicolour, and Virginica Type. Sepal length, Sepal width, Petal length, Petal width (numeric) and class(nominal).

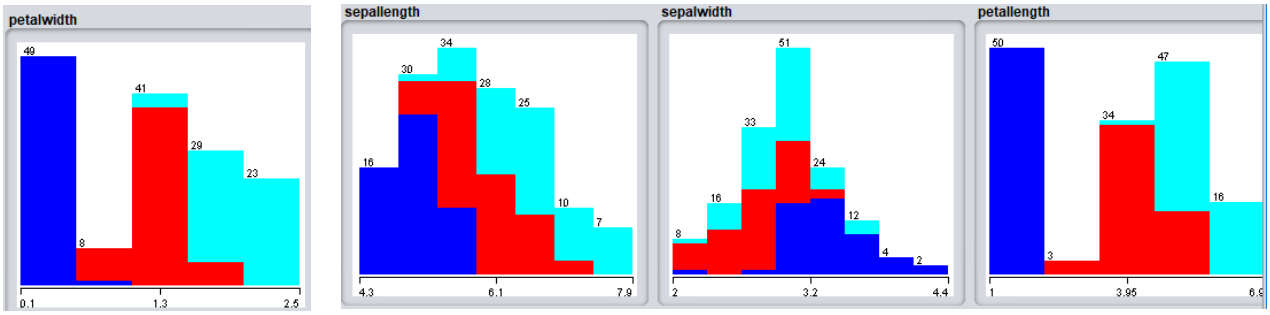
DIABETES : dataset consists of 768 people of having diabetes(positive or negative). preg, pres, skin, plas, insul, mass, pedi, age (numeric) and class (nominal)

Vote: consists of 425 people supporting democrats or republican and total of 17 attributes (Nominal)

1.2

Attribute	Min	Max	SD	Range
Sepal length	**4.3**	**7.9**	**0.828**	**4.3-7.9**
Sepal Width	**2**	**4.4**	**0.434**	**2-4.4**
Petal length	**1**	**6.9**	**1.764**	**1-6.9**
Petal Width	**0.1**	**2.5**	**0.763**	**0.1-2.5**

- Range is defined as the minimum and maximum value within which all the values are present.
- Range: Petal length > Sepal length and less range for sepal and petal width. Having a very narrow range won't be helpful as the classifier might not be able to distinguish the classes and to train, Also having a large range and if all the values are concentrated results in classification error.
- Iris dataset has varied ranges for all 4 attribute values. The values for sepal length and sepal width are distorted and not uniform for all classes. Whereas, for petallength and petalwidth, the values seem to be distributed evenly among all classes with proper distinction of attribute values to each of the classes in certain range. Petallength, followed by petalwidth attribute values have significance for deciding classes of the instances.
- According to results, for decision stump algorithm, petallength range of <=2.45 classifies as 'Iris-setosa' and >2.4 classifies as 'Iris-versicolor'. For random forest, the range for classification is <2.5-2.65 and >=2.5-2.65. For J48, range of petalwidth values decide the classification. Iris-setosa is classified in <=0.6 values whereas, Iris-versicolor and Iris-virginica are classified in >0.6 range values, which are further classified with new ranges of petallength and petalwidth.



- Degree of overlap is important for classifier as it helps the classifier to distinguish all the different classes. For the Iris dataset sepal length and width have larger overlap and classifier won't be able to distinguish which affects the performance, accuracy and time taken.
- Even though sepal length range is greater than petal width we can see that there is min overlap in petal width as all the values are concentrated in the sepal width (having a larger range is not enough, uniform distribution with minimum overlap is required).
- We can see that petal length has a large range and minimum overlap when compared to other attributes.

- From the above data, it can be inferred that using specific features like petal length and petal width, we can get a much better idea as to which class belongs to which range and thereby it becomes easy to differentiate the classes.

2.

There is a list of available parameters for the algorithm, which can be checked through Weka Explorer, below are some of important parameters

Decision Stump:

Parameter	Description	Analysis
batch Size	The preferred number of instances to process if batch prediction is being performed. More or fewer instances may be provided, but this gives implementations a chance to specify a preferred batch size. batch size is the number of training samples your training will use to make one update to the model parameters	batch size simply put, will simplify the process of updating the parameters. If the dataset has large no of instances, then giving batch Size will split the data into batches but the decision stump will give always give just 1 level of tree as output. If the data is small it is better to use the whole data but for a GB,TB of data it is better to have batches due to memory, performance and an advantage of keep updating the parameters

Random Forest:

Parameter	Description	Analysis
print Classifiers	Prints all the classifiers generated in Random Forest classifiers	Gives subtrees with varied ranges of attributes. Base tree remains the same. It helps a lot in visualizing the subtrees (each levels attribute).
numIterations	The number of iterations to be performed	Higher value will generate more trees. No. of trees come in output as per the value given. Must check the optimized no as it may take a lot of time with the larger amount of data
maxDepth	The maximum depth of the tree, 0 for unlimited	Changes to size of the tree. It helps to pre-prune the process and avoids overfitting. If it is 0 then the tree will be overfitted and must select a proper value so that it also doesn't lead to underfitting.

J48:

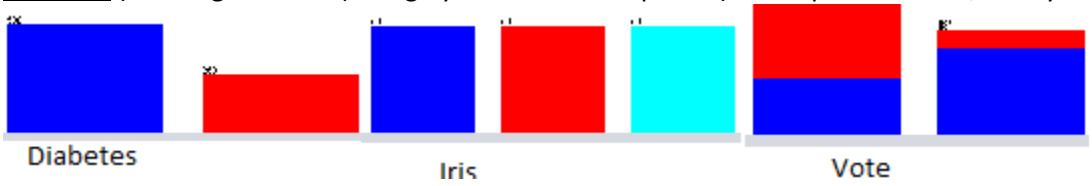
Parameter	Description	Analysis
minNumObj	The minimum number of instances per leaf.	No of leaves and nodes change. If it is increased/decreased error increases and size of tree increases/decreases. Increase of minimum number of objects per leaf causes the accuracy to drop down for few datasets and size, no of leaves decreases
confidenceFactor	The confidence factor used for pruning (smaller values incur more pruning).	Tree size and leaves are decreased for <0.25 with little effect on accuracy
Unprunned	set to either True or False for checking overfitting	This adds a check nodes per branches that can be removed which helps in reducing the overfitting with less effect on performance and time when set to false

Describe the class distribution for each dataset. How does it matter for the classifier?

IRIS : data is uniformly distributed for all three classes having 50 instances of each class (Iris Setosa, Iris versicolor and Iris virginica.)

Vote : number of instances for class democrat (267 instances) and is more than republicans(168 instances).

Diabetes (test-negative: 500) is highly skewed as compared (tested-positive : 268, nearly 50% of negative).



How classifier and data works

Decision Stump doesn't work very well with Iris dataset as it completely misclassify Iris virginica (it works best with two classes, decision stump uses only one attribute for making classification and gives only 1 level of tree in output), hence it works better on Vote dataset and Diabetes.

- J48, random forest works very well on IRIS,vote and diabetes

How data is affecting the performance of the classifier

- Attributes of diabetes have better distribution with lesser overlap when compared to the diabetes.
- The performance on diabetes dataset is less when compared to other datasets because the data is skewed. We can see that positive class is nearly half of negative class. Usually equal distribution is the best way for classifier than being skewed. The reason is that when we have lesser data for one of the classes the classifier becomes biased and tries to classify more samples to majority class. Steps to overcome this problem is
  1 Oversampling: try to increase the minority class samples by taking copies of these so that the ration is evenly distributed.
  2 Under sampling: try to decrease the majority class samples by taking copies of these so that the ration is evenly distributed, but we are losing data of much importance. Both are done on the training data
- Relative absolute error is greater for diabetes dataset due to these reasons when compared to other datasets

Class distribution for cross-validation

Use 10-fold cross-validation:

With 10-fold cross-validation, the original dataset is randomly partitioned into 10 equal size subsamples. Of the 10 subsamples, 1 is retained as the validation data for testing the model and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds can then be averaged to produce a single estimation. The advantage of this method is that all observations are used for both training and validation and each observation is used for validation exactly once.

Weka uses stratified cross-validation – that means that the class distribution in each train/test fold is the same as in the whole data set. Each class has about 33% of instances. It is very important to have stratified selection of examples when creating a training and test split, for cross-validation or for just using training/test set. It is necessary to train the classifier on the same class distribution as we have in the whole data and it makes sure all classes will be represented in the training and test sets.

Vote	Accuracy
Decision Stump	95.6
J48	96.7
Random Forest	96.8

Diabetes	Accuracy
Decision Stump	71.8
J48	75
Random Forest	75.3

IRIS	Accuracy
Decision Stump	66.6
J48	95.3
Random Forest	94.6

IRIS DATASET: Each training subset :9*15 = 135. Each Testing subset: 1 subset *15 = 15.
DIABETES DATASET. Each training :9 subsets*76 = 684 records. Size of the testing :1 subset* 84 =85
Vote: Each Training subset: 9*43 = 387records. Each testing 48 records.

10-fold cross validation:

In 10-fold cross validation the dataset is divided in to 10 equal parts and experiment is performed 10 times, each time 9 parts (90% of dataset) is used for training the model and 1 part (10%) is used for testing the model. Each part is hold out for testing in turns i.e. each part will be used to train the model 9 times and for testing the model 1 time. After the experiments is performed the average result of 10 experiments is taken in to consideration.

Class Distribution in 10-fold cross-validation

Weka by default uses Stratified cross-validation i.e. each folds (parts) contains equal proportion of class values. We can say that training and test set contains equal proportion of class values like original dataset. Stratified cross validation used by weka generates results with low variance as compared to normal cross validation techniques. It is extremely important as the algorithm generated will be less bias if each fold contains equal proportion of class values. It is better to use stratified

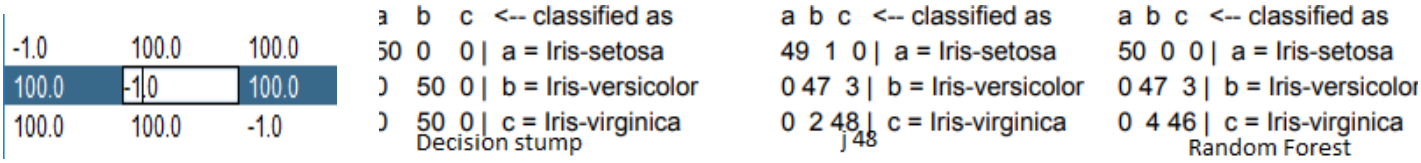
as this gives equal distribution like the original dataset. If each subset is selected randomly and dataset is skewed (one class has greater instances) then there might be probability of having very less minority class might be there

In the test options tab, choose "more options" and explore the classifier evaluation metric using a cost function. How did you set the cost parameter and why? How does it affect the results and why?

Steps:

- Weka Explorer→Classify tab→More Options→Enable 'Cost-sensitive evaluation'→Resize the class

Cost parameter is used to penalize the incorrect classifications like false positive and negative Using -1 for correct and penalty of 100. Used 10-fold cross validation



Decision Stump : (50 x -1) + (50 x -1) + (50 x 100) = 4900

J48: (49 x -1) + (1 x 100) + (47 x -1) + (3 x 100) + (2 x 100) + (-1 x 48) = 456

Random Forest: (50 x -1) + (47 x -1) + (46 x -1) + (3 x 100) + (4 x 100) = 557

IRIS DATASET	BEFORE		AFTER	
Decision Stump	Total Cost	50	Total Cost	4900
	Average Cost	0.33	Average Cost	32.3
J-48	Total Cost	6	Total Cost	456
	Average Cost	0.04	Average Cost	3.04
Random Forest	Total Cost	7	Total Cost	557
	Average Cost	0.05	Average Cost	3.7

Vote Dataset	BEFORE		AFTER	
Decision Stump	Total Cost	19	Total Cost	1990
	Average Cost	0.043	Average Cost	4.57
J-48	Total Cost	16	Total Cost	1669
	Average Cost	0.037	Average Cost	3.9
Random Forest	Total Cost	17	Total Cost	1800
	Average Cost	0.039	Average Cost	4.1

Diabetes Dataset	BEFORE		AFTER	
Decision Stump	Total Cost	216	Total Cost	21844
	Average Cost	0.28	Average Cost	28.442
J-48	Total Cost	201	Total Cost	19533
	Average Cost	0.26	Average Cost	25.4
Random Forest	Total Cost	186	Total Cost	18856
	Average Cost	0.24	Average Cost	24.5

- Decision stump and J48 gives higher costs because they incorrectly classify the instances, so the cost associated with them will be counted.
- Based on the cost for diabetes Random Forest performs well, Vote and Iris J-48 performs well. Thus, the cost parameter is used to impose penalty on misclassification.
- Thus cost parameter can be used to add penalty to the misclassification that are occurring and if the data is skewed and the minority classes are getting classified wrong, we can use the cost parameter to add penalty to FP,FN.

Describe the classification accuracy (on training and on test set, what is the difference? Does it matter?)

Used J48 classifier with different percentage split

Training data	66%	75%	85%	95%
Iris (Accuracy)	96	89.2	100	100
Diabetes(Accuracy)	76.24	77.08	78.26	78.95
Vote(Accuracy)	97.29	95.4	96.92	100

- We know that by 10-fold CV, Baseline accuracy (by default without any classifiers) for iris is 33.3%, vote is 61.38 % and Diabetes is 65%.
- Each classifier has accuracy greater than the baseline accuracy which shows that it is performing better than baseline accuracy.
- If we provide very small training data like 20% and test data of 80%, the classifiers accuracy is closer to baseline accuracy, which means that there is not enough data for the classifier to learn and build the model.
- Hence for the classifier to perform well it must be given certain amount of data else it leads to underfitting.
- If we try to give 95 and above percentage for training, in most of the cases it results in 100% accuracy for the classifiers. This shows that it leads to overfitting and there is no enough data for testing.
- Hence, we must select the training and test data such that there is no underfitting and overfitting.
- For smaller dataset, cross validation is better to perform, but for larger dataset it is better to use percentage split as CV takes lot of time on larger dataset. So, for larger dataset, it is optimal to use training data from 65 to 85% (must validate this, as the data is selected randomly for training and test data).
- As we have smaller dataset, we can see that it is optimal to use around 60 to 75 % training data (CV must be used for small dataset) to get an optimal classification accuracy.

Size of the tree, number of leaves:

Total no of leaves plus internal nodes forms the total size of the tree. No of leaves are the no of terminals / bottom layer of tree.

With J48:

	Size of the tree	Number of leaves
Iris	11	6
Vote	9	5
Diabetes	39	20

- Decision Stump will be only one level decision tree with root node connected to leaf nodes.
- Parameter like minNumObj, unpruned, depth, numFeatures ,numIterators and binarySplit have effect on tree size and number of leaves
- In Random Forest, the size and leaves of each tree is random (based on the features selected) but the no of trees can be controlled. Use numiterations and print the classifier to see the trees
- Below is a better explanation of the size and no of trees.

Parameter Analysis:

For J48 :minNumObj

	IRIS			Vote			Diabetes		
minNumObj	2	5	20	2	5	20	2	5	20
Accuracy	96	95.33	94	96.3	95.81	95.63	73.8281	74.6	74.87
Relative Error	7.87	10.241	13.82	12.887	13.98	18.84	69.5	69.5	69.95
No of leaves	5	4	3	6	5	2	20	13	10
Size of tree	9	7	5	11	9	3	39	25	19

Confidence Factor in J48:

	Vote			IRIS			Diabetes		
Confidence factor	0.05	0.25	0.4	0.05	0.25	0.4	.05	.25	0.4
Accuracy	96.09	96.32	96.32	94	96	96	74.5	73.82	72.9
Relative Error	13.83	12.887	12.8	11.5	7.87	7.87	71.0	69.348	69.03
No of leaves	6	6	6	4	5	5	12	20	22
Size of tree	11	11	11	7	9	9	23	39	43

Unpruned

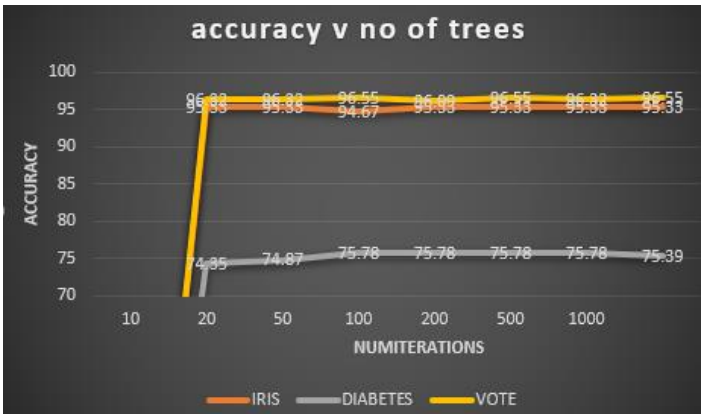
Dataset	Vote		IRIS		Diabetes	
unpruned	True	False	False	True	False	true
Accuracy	96.3	96.3	96	96	73.8	72.6
No of leaves	19	6	5	5	20	22
Size of tree	37	11	9	9	39	43

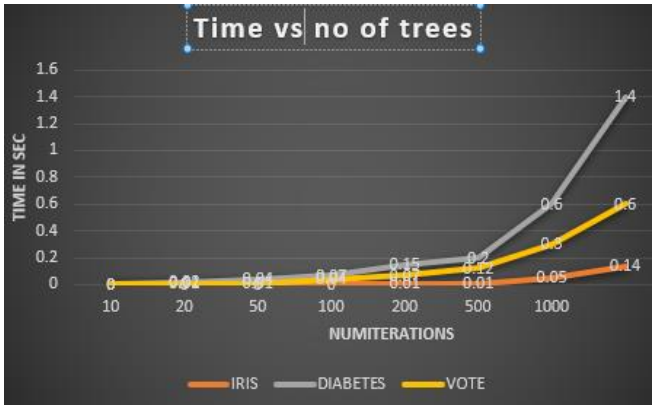
- **We can conclude that minNumobj increases, the size and the no of leaves decreases as we are increasing the min no of leaves per node, accuracy decreases with increase from 0.01 to 0.18 as we add more nodes which leads to error and decrease in accuracy.**
- **With the increase in the confidence factor the size and leaves increases with slight increase in accuracy**
- **Unpruned, accuracy remains same, but the size and leaves decrease if pruned which reduced the overfitting. This provides a better and noncomplex tree**

For the Random forest classifier analyze the number of trees. Find the parameter that you can use to set the number of trees.

numIterations is the parameter, which is used to set the number of trees in random forest.

	IRIS		**DIABETES**		**VOTE**	
No. of trees (numiterations)	**accuracy**	**Time in sec**	**Accuracy**	**Time in sec**	**Accuracy**	**Time in sec**
10	95.33	0.0	74.35	0.02	96.32	0.01
20	95.33	0.0	74.87	0.04	96.32	0.01
50	94.67	0.0	75.78	0.07	96.55	0.04
100	95.33	0.01	75.78	0.15	96.09	0.07
200	95.33	0.01	75.78	0.2	96.55	0.12
500	95.33	0.05	75.78	0.6	96.32	0.3
1000	95.33	0.14	75.39	1.4	96.55	0.6





- We can see from the plot that with the increase in the no of trees, the time taken for building the model increases along with the accuracy (there is a very small amount of effect). In general time taken to build increases with increase in the no of trees and the accuracy is also improved as there are more trees which improves the accuracy
- Processing time and performance both increase with increase in number of trees along with computations cost

Set the max depth of trees to 3 different values (small, medium, large). Use 3 different option for the number of trees (10, 100, 500).

maxDepth: defines complexity of the tree and performance. numIteration : defines the number of trees ,so cant use large no on a large dataset also the accuracy will be less if the no is less

		IRIS		Vote		Diabetes	
Max depth	**No of trees**	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**
0	**10**	**95.3**	**0**	**74.299**	**0.01**	**74.37**	**0.03**
	100	**95.3**	**0.02**	**79.9**	**0.08**	**75.78137**	**0.23**
	500	**95.3**	**0.08**	**80.37**	**0.39**	**75.78**	**1.34**
10	**10**	**95.3**	**0**	**75.7**	**0.01**	**75.13**	**0.02**
	100	**95.3**	**0.02**	**80.37**	**0.08**	**75.52**	**0.25**
	500	**95.3**	**0.09**	**80.37**	**0.5**	**75.78**	**1.3**
25	**10**	**95.3**	**0**	**74.29**	**0.01**	**74.35**	**0.03**
	100	**95.3**	**0.02**	**79.9**	**0.1**	**75.78**	**0.31**
	500	**95.3**	**0.06**	**80.37**	**0.46**	**75.78**	**1.2**

- We can conclude from the results that by keeping the max depth constant we can see with the increase in no of trees accuracy does increase along with more time taken to build the model. With more no of trees. Random forest classifier accuracy increases always but the time to taken to computer is high for larger dataset, hence it is recommended to have more no of trees along with consideration of the time taken for the datasets.
- By keeping no of trees same and changing the maxdepth, we can conclude that accuracy is maximum for optimal max depth, if max depth must be set such that it doesn't causes overfitting or underfitting.
- We can conclude that having more trees along with an optimal maxdepth is best as it increases the performance of the classifier
- Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. Random Forest prevents overfitting most of the time, by creating random subsets of the features and building smaller trees using these subsets. A more accurate prediction requires more trees, which results in a slower model.
- When printClassifier option is set to true, a chain of subtrees is generated on the console.
- All the subtrees that are generated are different with size and no of leaves.
- Each subtree doesn't have the same attribute at the node, after each iteration it tries to have better generalization with selection of different attribute (best).
- When tried for Iris dataset and setting no of trees to 5 we see that only petal length is at the root of the subtree, but when we increase it to 100 we can see petal width selected at the root for some of the subtrees.

- Also, the size of each tree is different, which shows that why random forest performs better as it takes the average of all the trees.
- The accuracy of the classifier also increased with more no of trees (for no of trees =2 accuracy is 93 and no of trees is 2000 accuracy is 95.33)
- This showed that random forest performs very well with large no of trees and it is always recommended to use it, the one drawback is that it is not recommended for real time as it takes more time

3. Feature Selection:

CorrelationAttributeEval: based on the correlation between the attribute and class it provides the rank for the attributes
InfoGainAttributeEval: based on the Information gain of the attribute, it provides the rank for the attributes.
Below are the features ranking based on correlation and information gain on IRIS, Vote and Diabetes dataset.

CorrelationAttributeEval:	InfoGainAttributeEval:
0.615 3 petallength	1.418 3 petallength
0.592 4 petalwidth	1.378 4 petalwidth
0.478 1 sepallength	0.698 1 sepallength
0.397 2 sepalwidth	0.376 2 sepalwidth

CorrelationAttributeEval	InfoGainAttributeEval:
0.4666 2 plas	0.1901 2 plas
0.2927 6 mass	0.0749 6 mass
0.2384 8 age	0.0725 8 age
0.2219 1 preg	0.0595 5 insu
0.1738 7 pedi	0.0443 4 skin
0.1305 5 insu	0.0392 1 preg
0.0748 4 skin	0.0208 7 pedi
0.0651 3 pres	0.014 3 pres

CorrelationAttributeEval	InfoGainAttributeEval
0.9096 4 physician-fee-freeze	0.7078541 4 physician-fee-freeze
0.7343 3 adoption-of-the-budget-res	0.4185726 3 adoption-of-the-budget-res
0.6837 5 el-salvador-aid	0.4028397 5 el-salvador-aid
0.6666 12 education-spending	0.34036 12 education-spending
0.6283 9 mx-missile	0.3123121 14 crime
0.617 8 aid-to-nicaraguan-contras	0.3095576 8 aid-to-nicaraguan-contras
0.6063 14 crime	0.2856444 9 mx-missile
0.5268 13 superfund-right-to-sue	0.2121705 13 superfund-right-to-sue
0.5127 15 duty-free-exports	0.2013666 15 duty-free-exports
0.5045 7 anti-satellite-test-ban	0.1902427 7 anti-satellite-test-ban
0.413 6 religious-groups-in-schools	0.1404643 6 religious-groups-in-schools
0.3931 1 handicapped-infants	0.1211834 1 handicapped-infants
0.3669 11 synfuels-corporation-cutbac	0.1007458 11 synfuels-corporation-cut
0.3519 16 export-administration-act-SA	0.0529956 16export-administration-actSA
0.0838 10 immigration	0.0049097 10 immigration
0.011 2 water-project-cost-sharing	0.0000117 2 water-project-cost-sharing

- petallength' is the attribute I had used during the data analysis, which is the same after ranked from above feature selection. We can classify instances using the petal length range values.
- 'physian-fee-freeze' can classify between democrat and republican with counts of yes/no votes. Also, it has the highest information gain so, it holds more information to classify number of votes.
- 'plas' can classify between tested negative and tested positive with counts of yes/no. Also, it has the highest information gain so, it holds more information to classify number of votes.
- If the dataset has less no of attributes, in which almost all are used to decide the classification of the instances then if some attributes are removed then there will not be any performance impact. But, if there are more no of attributes in the dataset then, removing lower ranked attributes will improve the performance.
- We can see that petal length is selected in the iris dataset, vote dataset plas is selected and then mass, Diabetes physician fee freeze is selected.
- feature selection importance.
  - **Reduces Overfitting:** Many features and low samples/features ratio will introduce noise which leads to overfitting and false performance
  - **Reduces Training Time and Improves Accuracy:** Reducing the number of features will reduce the running time which also helps you to select higher complex algorithms.

- Some algorithms would work better on some set of features while some other algorithms on other.
- Thus, by selecting less no of features that capture all the variance. helps to improve performance.
- There are also other methods like PCA for dimensionality reduction

4. Noise and Missing values

Missing values:

For Iris dataset

According to results, as the amount of missing values introduced increases in the data set, percentage of accuracy decreases, percentage of error increases and the time for training model increases for all classifiers

	Decision stump		Random forest		J48	
	Accuracy (%)	Time in sec	Accuracy (%)	Time in sec	Accuracy (%)	Time in sec
Default	66.67	0.01	100	0	98	.01
5%	66.67	0.01	94.67	0	96.15	0.01
10%	64	0.01	93.2	0.01	94.6	0.1
50%	52	0.01	94.3	0.02	93.4	0.2

Noise values:

For iris dataset,

- According to results, random forest classifier provides comparatively good accuracy than other two classifiers.
- If percentage of noise increases, accuracy decreases, error increases and the time for training model increases.

	Decision stump		Random forest		J48	
	Accuracy (%)	Time in sec	Accuracy (%)	Time in sec	Accuracy (%)	Time in sec
Default	66.67	0	100	0	98	0.01
5%	65.1	0	94.67	0	91.2	0.02
10%	63.08	0	93.9	0.01	86.58	0.1
50%	66.42	0.01	94.1	0.01	65.08	0.2

Inorder for the performance to improve, We usually use mean or median to replace all the null values

Change the range of one attribute, make it 1000 times larger than the other features range. How does this affect the classifier's accuracy?

Step : randomly choose any one attribute ,multiply 1000 to all the instances of that attribute value

	Decision Stump	J48	Random Forest
Sepal length	**66.6**	**96**	**95.3**
Petal length	**66.6**	**96**	**95.3**
Sepal width	**66.6**	**96**	**95.3**
Petal width	**66.6**	**96**	**95.3**

- If the range of an attribute is increased to 1000 times, still the accuracy remains unchanged. It has no effect on accuracy

Conclusion

- Range of an attribute is the measure of spread dispersion of a value indicates whether the attribute values are widely spread or relatively concentrated around a single point. Having a large range or low range doesn't make the attribute to the best for classifier. Even after having large range if we have no uniform distribution (overlapping) with values concentrated, then we will select an attribute with lesser range that has minimum overlapping with proper distribution of classes.

- Hyperparameters: Each classifier has its own parameters that has its own importance and plays an important role in improving classifier accuracy
- Ideal dataset Is to have equal distribution of the classes, if the data is skewed then it will affect the classifier performance. The minority classes will not be classified correctly and results in decrease in accuracy. Few methods employed for this is under sampling and oversampling.
- If the data is large it is ideal to use the percentage split but for smaller datasets it is recommended to use cross validation.
- Rather than random selection of classes for each fold in cross validation it is better to have uniform distribution like original dataset because during random selection there is a probability of having very less samples of minority class which decreases the performance of classifier.
- WEKA uses stratified cross-validation – that means that the class distribution in each train/test fold is the same as in the whole data set. It is important to train the classifier on the same class distribution as we have in the whole data and it makes sure all classes will be represented in the training and test sets.
- Cost parameter can be used to add a penalty to the misclassification Example: if the patient having a cancer is not diagnosed, this will be a problem in the real world. hence, we can use the cost parameter to add penalty to FP and FN to check misclassification.
- Regarding classification accuracy, we can conclude that setting the percentage split around 65% to 85% (which varies for data set) the classifier has highest percentage of correctly classified instances. Hence at times an optimal training and test data must be used inorder to remove underfitting, overfitting such that it improves the performance of the classifier.
- Time taken to build the model increases as the no of trees increases as the structure becomes more and more complex the time to build is greater.
- With increase in no of trees, performance of the Random Forest classifier increases, hence it is recommended to use a higher no of trees with optimal time for building the model. Random Forest is not the best for real time analysis as it takes more time
- For feature selection, attribute with the highest rank is the best for classification and the one with the least rank contributes less and when removed there is very slight change in the performance and decrease in time required. With the proper knowledge of the domain and feature selection, we can select the features that are required. Hence when we remove the least ranked attributes and the performance remains unchanged, this parameter allows us to achieve dimensionality reduction. PCA is another method to achieve dimensionality reduction
- Having an optimal maxdepth and more no of trees such that it doesn't take long time would be the best for the classifier
- For small missing value percentage, the performance is not affected lot, but with larger percentage of missing values the accuracy of the classifier decreases, and time taken also increases.
- With increase in number of noises values the algorithm performance of algorithm decreases but if instead a default value is provided then performance improves.
- We usually use mean or median to replace all the null values.
- Decision stump works well on binary classification and if the attribute has stringer relation with the class.
- Random Forest provides better performance but takes time and is more complex when compared to J48