Applying Different Data Mining Approach to Dress Sales Recommendation using Weka

By Achyut Saroch RA1811003030302

Abstract :

Data Mining is defined as a process used to extract usable data from a larger set of any raw data. It implies analysing data patterns in large batches of data using one or more software. As an application of data mining, businesses can learn more about their customers and develop more effective strategies related to various business functions and in turn leverage resources in a more optimal and insightful manner.In this work we give the essential Data Mining step such as preprocessing data (remove outlier, replacing missing value etc), attribute selection, aim to choose just relevant attributes and remove the irrelevant attribute and redundant attribute, classification and assessment of varied classifier model. The Weka small DataMining Tool that is popular in beginner and research. This tool consists of a lot of algorithms for Attribute Selection, Classification Regression and Classification.

1. Introduction :

Data Mining is a process of discovering interesting patterns and knowledge from large amounts of data. The data sources can be anything(CSV file , Web Page, Data Warehouse, DataSet hosting Website).Data mining involves effective data collection and warehousing as well as computer processing. For segmenting the data and evaluating the probability of future events, data mining uses sophisticated mathematical algorithms. Data mining is also known as Knowledge Discovery in Data (KDD).Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.
Classification methods divided into: unsupervised or supervised. In supervised classification, one attribute of the dataset includes predetermined values that represent a collection of the data. These collections are

called classes. For unsupervised classification the objective is partition into groups or clusters.Then the observations of the dataset is based on some logical relationship that exists among the values of the attributes but that must yet be discovered.

Waikato Environment for Knowledge Analysis (Weka), developed at the University of Waikato, New Zealand, is free software licensed under the GNU General Public License, and the companion software to the book "Data Mining: Practical Machine Learning Tools and Techniques".Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

2. DESCRIPTION OF DATASET

The dataset represents dress and sales and recommendation according to the sales. The dataset includes 13 features and a class.
Link of DataSet : https://www.openml.org/d/1470 (Data set was originally from UCI but I downloaded from openml.org it is the same data but not with feature's name and missing Dress_Id which I have added on my own. Original link to data : https://archive.ics.uci.edu/ml/datasets/dresses_attribute_sales)
DataSet was prepared by Muhammad Usman & Adeel Ahmed from Air University Islamabad

DataSet Information:
Dress_id, Style, Price, Rating, Size, Season, NeckLine, SleeveLength, waiseline, Material, FabricType, Decoration, Pattern, Type, Recommendation are Attributes in the dataset.

Attribute Information:
Dress_id: digit (this is added column in data, original data didn't contain Dress_id)
Style: Bohemia,brief,casual,cute,fashion,flare,novelty,OL,party,sexy,vintage,work.
Price:Low,Average,Medium,High,Very-High
Rating:1-5
Size:S,M,L,XL,Free
Season:Autumn,winter,Spring,Summer
NeckLine:O-neck,backless,board-neck,Bowneck,halter,mandarin-collor,open,peterpan-collor,ruffled,scoop,slash-neck,square-collar,sweetheart,turndowncollar,V-neck.
SleeveLength:full,half,halfsleeves,butterfly,sleveless,short,threequarter,turndown,null
waistline:dropped,empire,natural,princess,null
Material:wool,cotton,mix etc
FabricType:shafoon,dobby,popline,satin,knitted,jersey,flannel,corduroy etc
Decoration:applique,beading,bow,button,cascading,crystal,draped,embroridary,feathers,flowers etc
Pattern type: solid,animal,dot,leopard etc
Recommendation:0,1

No.		Name
1		Dress_id
2		Style
3		Price
4		Rating
5		Size
6		Season
7		NeckLine
8		SleeveLength
9		Waistline
10		Material
11		FabricType
12		Decoration
13		PatternType
14		Recommendation

fig1: List of Attributes

File Edit View

Attribute DataSet - Sheet1.csv

Relation: Attribute DataSet - Sheet1

No.	1: Dress_id Numeric	2: Style Nominal	3: Price Nominal	4: Rating Numeric	5: Size Nominal	6: Season Nominal	7: NeckLine Nominal	8: SleeveLength Nominal	9: Waistline Nominal	10: Material Nominal	11: FabricType Nominal	12: Decoration Nominal	13: PatternType Nominal	14
1	1.0060...	Sexy	Low	4.6	M	Summer	o-neck	sleevless	empire	null	chiffon	ruffles	animal	
2	1.2121...	Cas...	Low	0.0	L	Summer	o-neck	Petal	natural	microfiber	null	ruffles	animal	
3	1.1903...	vint...	High	0.0	L	Automn	o-neck	full	natural	polyster	null	null	print	
4	9.6600...	Brief	Aver...	4.6	L	Spring	o-neck	full	natural	silk	chiffon	embroidary	print	
5	8.7633...	cute	Low	4.5	M	Summer	o-neck	butterfly	natural	chiffonfa...	chiffon	bow	dot	
6	1.0683...	boh...	Low	0.0	M	Summer	v-neck	sleevless	empire	null	null	null	print	
7	1.2207...	Cas...	Aver...	0.0	XL	Summer	o-neck	full	null	cotton	null	null	solid	
8	1.2196...	Nov...	Aver...	0.0	free	Automn	o-neck	short	natural	polyster	broadcloth	lace	null	
9	1.1130...	Flare	Aver...	0.0	free	Spring	v-neck	short	empire	cotton	broadcloth	beading	solid	
...	9.8529...	boh...	Low	0.0	free	Summer	v-neck	sleevless	natural	nylon	chiffon	null	null	
...	1.1172...	party	Aver...	5.0	free	Summer	o-neck	full	natural	polyster	broadcloth	lace	solid	
...	8.9848...	Flare	Aver...	0.0	free	Spring	v-neck	short	null	nylon	null	null	animal	
...	9.5772...	sexy	Low	4.7	M	Winter	o-neck	threequarter	null	null	chiffon	lace	print	
...	7.4903...	vint...	Aver...	4.8	M	Summer	o-neck	short	empire	cotton	jersey	null	animal	
...	1.0554...	Cas...	Low	5.0	M	Summer	boat-neck	short	null	cotton	null	sashes	solid	
...	1.1626...	Cas...	Low	0.0	free	Winter	boat-neck	full	null	other	other	lace	null	
...	6.2431...	cute	Aver...	4.7	L	spring	o-neck	short	null	cotton	null	sashes	solid	
...	8.3046...	boh...	Med...	5.0	free	Automn	o-neck	full	natural	null	null	hollowout	patchwork	
...	8.4085...	Brief	Aver...	0.0	M	Winter	peterpa...	threequarter	natural	cotton	null	null	patchwork	
...	1.1132...	Sexy	Aver...	5.0	M	Automn	o-neck	sleevless	empire	milksilk	null	null	null	
...	8.6175...	Sexy	Aver...	4.5	L	Automn	o-neck	full	null	cotton	null	beading	solid	
...	8.5617...	Cas...	Low	4.3	M	Summer	o-neck	sleevless	natural	null	chiffon	null	solid	
...	1.1229...	Brief	Low	4.0	XL	Summer	v-neck	short	natural	cotton	null	pockets	solid	
...	8.4051...	Sexy	Aver...	4.7	S	Summer	v-neck	sleevless	empire	cotton	null	sequined	solid	
...	7.6851...	Sexy	Aver...	0.0	free	Automn	v-neck	sleevless	natural	polyster	null	lace	patchwork	
...	1.1398...	Sexy	Aver...	0.0	M	Automn	o-neck	sleevless	empire	co[illegible]	[illegible]	[illegible]	print	
...	1.0042...	Sexy	Aver...	4.7	M	Spring	o-neck	sleevless	natural	null	null	null	solid	
...	1.2354...	Cas...	Low	0.0	L	Summer	o-neck	sleevless	natural	cotton	jersey	null	print	
...	9.4280...	cute	Low	4.3	free	Automn	o-neck	sleevless	natural	polyster	chiffon	sashes	striped	
...	6.2913...	cute	Low	4.7	M	Spring	ruffled	short	empire	chiffonfa...	chiffon	bow	dot	
...	8.5194...	Cas...	Aver...	4.6	L	Automn	o-neck	sleevless	empire	polyster	chiffon	ruffles	solid	
...	1.1502...	Cas...	Low	0.0	M	Spring	o-neck	sleevless	natural	silk	chiffon	applique	solid	
...	1.0266...	Cas...	Aver...	4.4	L	Automn	o-neck	sleevless	natural	linen	chiffon	ruffles	animal	
...	9.7877...	Brief	Aver...	4.3	L	Spring	o-neck	sleevless	empire	null	null	bow	solid	
...	8.2779...	party	High	4.7	M	Spring	o-neck	threequarter	null	null	null	null	print	
...	6.4082...	Cas...	Low	4.6	free	Winter	boat-neck	full	natural	cotton	null	null	patchwork	
...	8.0186...	vint...	Aver...	4.6	free	Winter	o-neck	full	natural	null	chiffon	null	print	
...	1.0602...	Brief	Aver...	0.0	M	Automn	o-neck	short	natural	cotton	null	null	print	
...	1.0546...	party	Aver...	4.6	M	Winter	o-neck	sleevless	natural	cotton	null	lace	solid	

Right click (or left+alt) for context menu

Fig2: Raw data

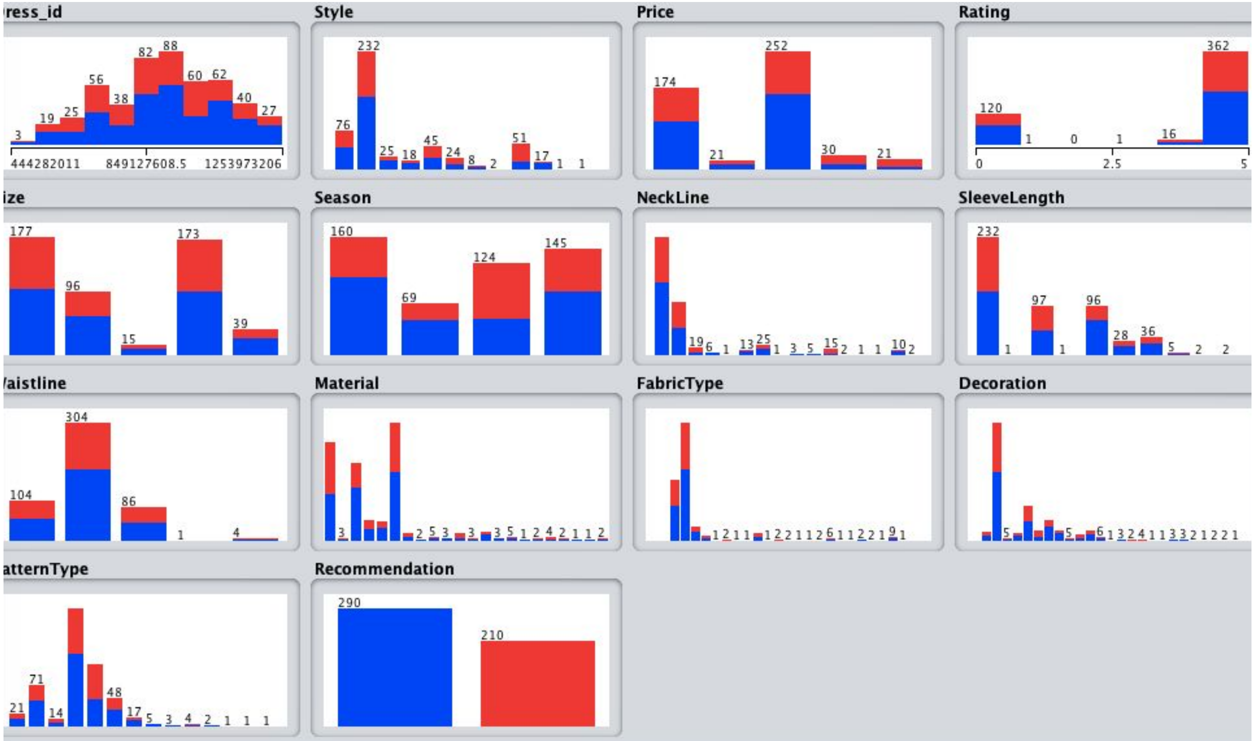


Fig3: Attribute Visualization

3. Pre Examination of Data:

1.There is no obvious pattern in the dataset.

2.There are missing value in attributes

SleeveLength	Waistline	Material	FabricType
sleevless	empire	null	chiffon
Petal	natural	microfiber	null
full	natural	polyster	null
full	natural	silk	chiffon
butterfly	natural	chiffonfabric	chiffon
sleevless	empire	null	null
full	null	cotton	null
short	natural	polyster	broadcloth
short	empire	cotton	broadcloth
sleevless	natural	nylon	chiffon
full	natural	polyster	broadcloth
short	null	nylon	null
threequarter	null	null	chiffon
short	empire	cotton	jersey

Fig4: showing missing (null) data

4. Material and Method:

4.1 DataCleaning

DataCleaning is necessary to remove incomplete and inconsistent data from the DataSet. Like in feature Season we have duplicate attribute:

Name: Season | Type: Nominal
Missing: 2 (0%) | Distinct: 8 | Unique: 1 (0%)

No.	Label	Count	Weight
1	Summer	159	159.0
2	Automn	61	61.0
3	Spring	122	122.0
4	Winter	99	99.0
5	spring	2	2.0
6	winter	46	46.0
7	summer	1	1.0
8	Autumn	8	8.0

In the Above Image [{Spring,spring},{Winter,winter},{Summer,summer},{Automn,Autumn}] are the same Attribute.

The only option is to replace this duplicate Attribute in the .arff file of the data.We can do this in any editor but for this experiment I am using SublimeText.

```
@attribute Season {'?',Automn,Autumn,spring,Spring,summer,Summer,winter,Winter}
```

First remove the duplicate from "@attribute Season" and then in SublimeText open Find and Replace all value of Autom with Autumn,

Name: Season | Type: Nominal
Missing: 2 (0%) | Distinct: 8 | Unique: 1 (0%)

No.	Label	Count	Weight
1	Summer	159	159.0
2	Automn	61	61.0
3	Spring	122	122.0
4	Winter	99	99.0
5	spring	2	2.0
6	winter	46	46.0
7	summer	1	1.0
8	Autumn	8	8.0

fig5:Before Cleaning DataSet

Name: Season | Type: Nominal
Missing: 2 (0%) | Distinct: 4 | Unique: 0 (0%)

No.	Label	Count	Weight
1	Summer	160	160.0
2	Autumn	69	69.0
3	Spring	124	124.0
4	Winter	145	145.0

fig6:After Cleaning DataSet

We have the same problem with Attribute: NeckLine,Style,Size,Price,SleevesLength,Material this can be solved as the same way above.

For removal of **Noise (**contains error and outliers **)** from the data, First we have to check for outliers in our DataSet this can be done by using Weka in build feature **InterquartileRange,** this filter adds new attribute that point out weather values for instances can be considered outliers or extreme values.

The outliers can be removed by using Weka inBuild **RomoveWithValues** in unsupervised instances.(outliers in this data set were 124 total data left are 376)

Choose->filters->unsupervised->instance->RemoveWithValuesset the properties of attributIndex to 14(the index of outlier attribute) and nominalIndices to last(to remove just the last values that have the value yes) and after applying the result is

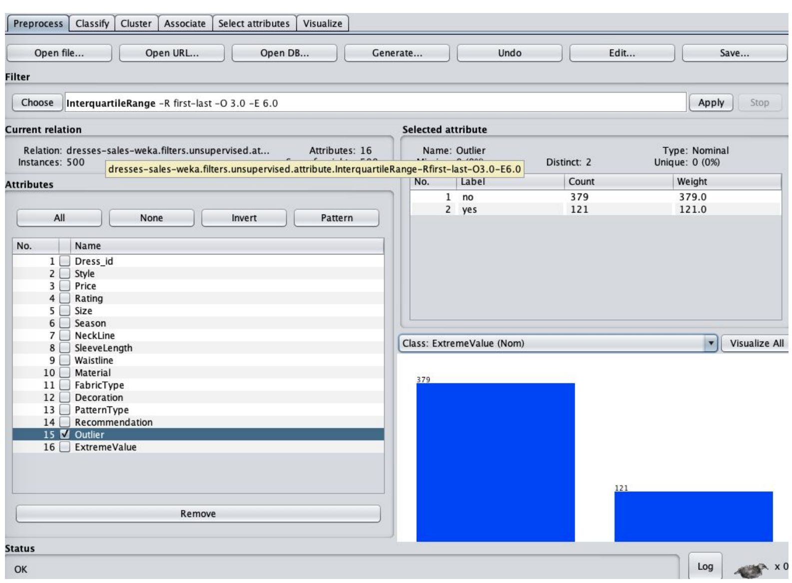


Fig7: Before removing outliers

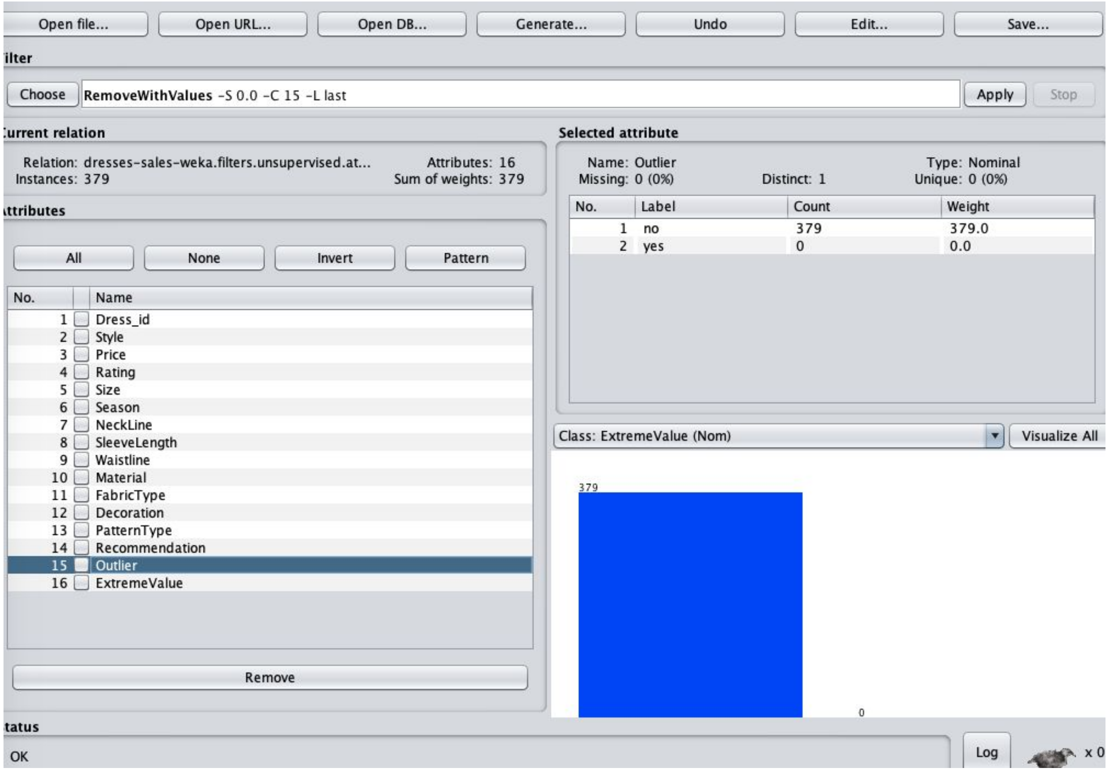


Fig8: After removing outliers

4.2 Data Reduction:

In this technique we make a new data representation that is considerably smaller in magnitude but so far produces the same analytical result.

Numerosity reduction : we can be applied Numerosity reduction technique to decrease the data volume by choosing alternative, smaller forms of data representation.

For example in Feature Season we have 4 attribute

{Summer,Autumn,Spring,Winter} --> {1,2,3,4}

Some values have a large scale so we can automatically normalize the data by using WEKA filter discretizes.

Dimensionality reduction: Dimensionality reduction reduces the dataset volume by removing irrelevant attributes.

Here we use the filter AttributeSelection:

Choose->filters->supervised->attribute->AttributeSelection And the result is:

No.		Name
1		Dress_id
2		Style
3		Price
4		Rating
5		Size
6		Season
7		NeckLine
8		SleeveLength
9		Waistline
10		Material
11		FabricType
12		Decoration
13		PatternType
14		Recommendation
15		Outlier
16		ExtremeValue

Fig9: Before dimension reduction

No.		Name
1		Style
2		Price
3		Season
4		NeckLine
5		SleeveLength
6		Material
7		FabricType
8		PatternType
9		Recommendation

Fig10: After dimension reduction

5. Classification:

Classification could be a data processing function that assigns items during a collection to focus on categories or classes. The goal of classification is to accurately predict the target class for every case within the data. For instance, a classification model may well be accustomed to identify loan applicants as low, medium, or high credit risks.A classification task begins with an information set within which the category assignments are known. As an example, a classification model that predicts credit risk may be developed to support observed data for several loan applicants over a period of your time. Additionally to the historical credit rating, the info might track employment history, home ownership or rental, years of residence, number and kind of investments, and so on. Credit rating would be the target, the opposite attributes would be the predictors, and therefore the data for every customer would constitute a case.Classifications are discrete and don't imply order. Continuous, floating-point values would indicate a numerical, instead of a categorical, target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm.

The simplest style of classification problem is binary classification. In binary classification(Classification where only we have to divide the DataSet into two Clusters), the target attribute has only two possible values: for instance, high credit rating or low credit rating. Multiclass targets have quite two values: as an example, low, medium, high, or unknown credit rating.In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and therefore the values of the target. Different classification algorithms use different techniques for locating relationships. These relationships are summarized in an exceedingly model, which might then be applied to a unique data set within which the category assignments are unknown.Classification models are tested by comparing the expected values to known target values in a very set of test data. The historical data for a classification project is usually divided into two data sets: one for building the model; the opposite for testing the model.

There are many type of Data Mining Algorithm but in this Project I'll use:

1. Decision Tree
2. Naive Bayes

5.1 : Decision Tree

Decision Tree is a tree shaped diagram used to determine a course of action. Each branch of the tree illustrates a possible decision, occurrence of reaction. Decision Tree Solve both Classification and Regression Problems. In Classification, A Classification tree will determine a set of logical if-then conditions to classify problems.For example. Discriminating between three types of flowers based on certain features.In Regression, A Regression Tree is used when the target variable is numerical or continuous in nature. We fit a regression model to the target variable using each of the independent variables. Each split is made based on the sum of squared error.

Advantages:

1. Simple to understand, interpret and visualize.
2. Little effort is required in data preparation.
3. Can handle both numerical and categorical data
4. Nonlinear parameters don’t affect the performance.

Disadvantages:

1. Overfitting occurs when the algorithm captures noise in the data.
2. Models can get unstable to small variations in data.

3. A highly complicated Decision tree tends to have a low bias which makes it difficult for the model to work with new data.

Important term in Decision Tree:

1. Entropy: The measure of randomness or unpredictability in the DataSet. Example {1,2,2.34,-12,2,4/3,9}
2. Information Gain: The measure of decrease in entropy after the DataSet is split.
3. LeafNode: Node carries the Classification or the Decision.
4. RootNode: The top most Decision Node

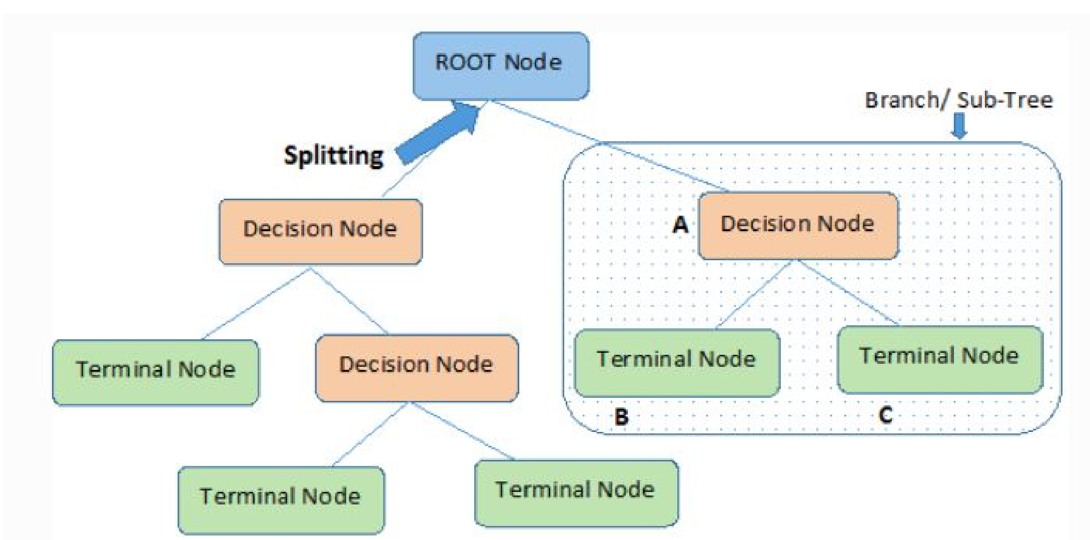
Working of Decision Tree:

The main part of the algo is to separate the information.In decision tree we've to separate data in such the simplest way that the data gain is the highest(to decrease this entropy). therefore the Entropy is define by the Formula :

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Where ‘pi’ solely represents the frequentist probability of an element/class ‘i’ in our data. For example let’s say we only have two classes , a positive class and a negative class. Therefore ‘i’ here can be either + or (-). So if we had a complete of 100 data points in our dataset with 30 belonging to the positive class and 70 belonging to the negative class then ‘p+’ would be 0.3 (3/10) and ‘p-’ would be 0.7 (7/10).

This Splitting work till E(s) = 0.



```
INPUT: S, where S = set of classified instances
OUTPUT: Decision Tree
Require: S ≠ ∅, num_attributes > 0
procedure BUILDTREE
    repeat
        maxGain ← 0
        splitA ← null
        e ← Entropy(Attributes)
        for all Attributes a in S do
            gain ← InformationGain(a, e)
            if gain > maxGain then
                maxGain ← gain
                splitA ← a
            end if
        end for
        Partition(S, splitA)
    until all partitions processed
end procedure
```

Fig11: Decision Tree Algo (PseudoCode)

In Weka the Decision Tree is under the name of REPTREE in the classifiers.

5.2 Naive Bayes:

Naive Bayes Classifier works on the principles of conditional probability as given by the Bayes Theorem. Bayes' Theorem gives the conditional probability of an event A given another event B has occurred.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B	= events
$P(A\|B)$	= probability of A given B is true
$P(B\|A)$	= probability of B given A is true
$P(A), P(B)$	= the independent probabilities of A and B

Naive Bayes only work for classification of Dataset, don't work on regression.

Advantages:

1. Naive Bayes is easy and fast to predict class(both single and multi-class) of test data sets.
2. In assumption of independence Naive Bayes classifier performs better compared to other models like logistic regression.
3. It performs well in case of categorical input variables compared to numerical variable(s).

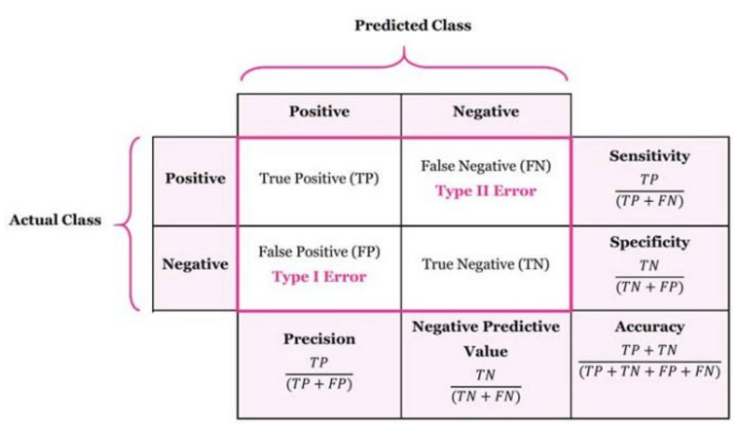
Disadvantage:

1. If a categorical variable has a category (in the test data set), which was not observed in the training data set, then the model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
2. Naive Bayes is also known as a bad estimator.
3. Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

In Weka the Naive Bayes Classifier is under the name of NaiveByes in the classifiers.

5.3 Precision,Sensitivity,Specificity and Accuracy.

Precision ,Sensitivity, Specification and Accuracy are the property that tell how good the Algorithm works on the DataSet in classification.In this project I will be using Accuracy and Sensitivity (Recal)l to Calculate the F1-score of the model working on the DataSet that will Tell us which model is better on DataSet.



Definition of the Term:

1. True Positive (TP) : Observation is positive, and is predicted to be positive.
2. False Negative (FN) : Observation is positive, but is predicted negative.
3. True Negative (TN) : Observation is negative, and is predicted to be negative.
4. False Positive (FP) : Observation is negative, but is predicted positive.

Precision => TP(TP + FP)

Sensitivity(recall) => TP/(TP+FP)

Specificity => TN/(TN + FP)

Accuracy => (TP + TN)/(TP + TN + FP +FN)

F1-Score => 1/2*((Precision*Recall)/(Precision + Recall))

6. Result

I apply **Decision Tree** algorithm with 5 different models of dividing the dataset to training and testing Ranging from (90:10) to (60:40) in given below table:

9-Attribute				
Model num	DataAllocation	Precision	F-Score	MeanSqauredError
1	90/10	0.506	0.503	0.5704
2	80/20	0.444	0.445	0.5569
3	70/30	0.696	0.5	0.4879
4	60/40	0.599	0.749	0.4925
5	66/34	0.586	0.569	0.5438
12-Attribute				
Model num	DataAllocation	Precision	F-Score	MeanSqauredError
1	90/10	0.547	0.55	0.5376
2	80/20	0.508	0.501	0.5808
3	70/30	0.649	0.645	0.5534
4	60/40	0.505	0.51	0.5641
5	66/34	0.468	0.474	0.6067

Fig12: DT F-Score for 5 models with 9 Attribute and with 12 Attribute

I apply **Naive Bayes** algorithm with 5 different models of dividing the dataset to training and testing Ranging from (90:10) to (60:40) in given below table:

9-Attribute				
Model num	DataAllocation	Precision	F-Score	MeanSqauredError
1	90/10	0.547	0.55	0.5305
2	80/20	0.587	0.571	0.5117
3	70/30	0.673	0.668	0.476
4	60/40	0.637	0.638	0.4805
5	66/34	0.593	0.594	0.4894
12-Attribute				
Model num	DataAllocation	Precision	F-Score	MeanSqauredError
1	90/10	0.515	0.519	0.5483
2	80/20	0.51	0.505	0.5343
3	70/30	0.609	0.61	0.5014
4	60/40	0.6	0.599	0.507
5	66/34	0.619	0.62	0.5149

Fig13: NB F-Score for 5 models with 9 Attribute and with 12 Attribute

Model name	DT F-score	NB F-Score
model1_90	0.503	0.55
model2_80	0.445	0.571
model3_70	0.5	0.668
model4_60	0.749	0.638
model5_66	0.569	0.594

Fig14: comparison between DT and NB F-score

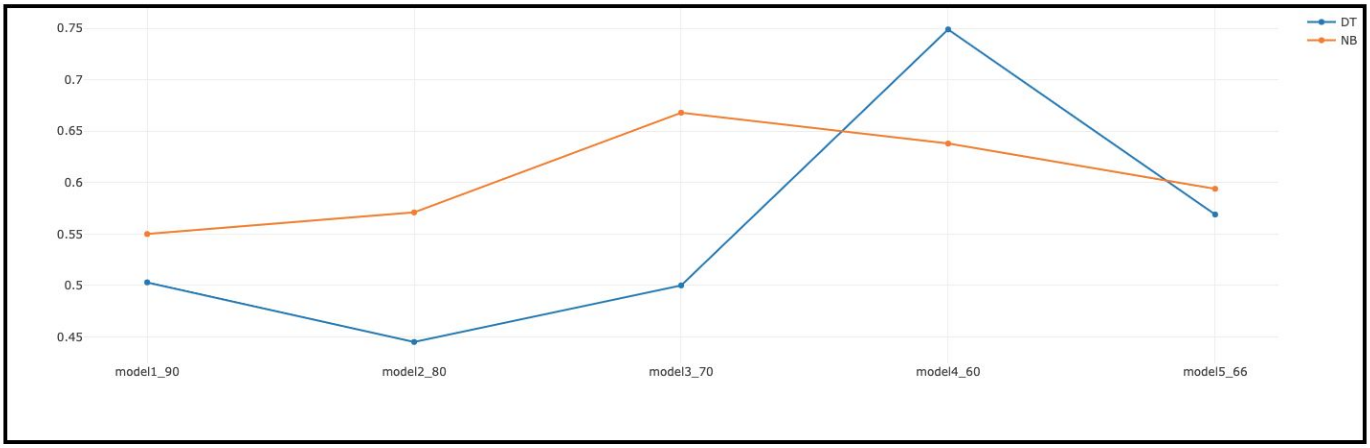


Fig14. Comparative of F-score results for DT and NB

7.Conclusion :

DataSet was originally from UCI (https://archive.ics.uci.edu/ml/datasets.php), firstly we preprocess the dataset using Weka then we put the resulting preprocessed data in classification models for two algorithms (DT, NB). We can also see that there is a difference in accuracy when the model used 9 features which were selected by AttributeSelection in Weka unlike the original 12 features.Then we examined the best performance between the two classifiers model by comparative the F-score and meanSqaueredError using the dataset after applying AttributeSelection. And we can see that Naive Bayes is performing much better than Decision tree.