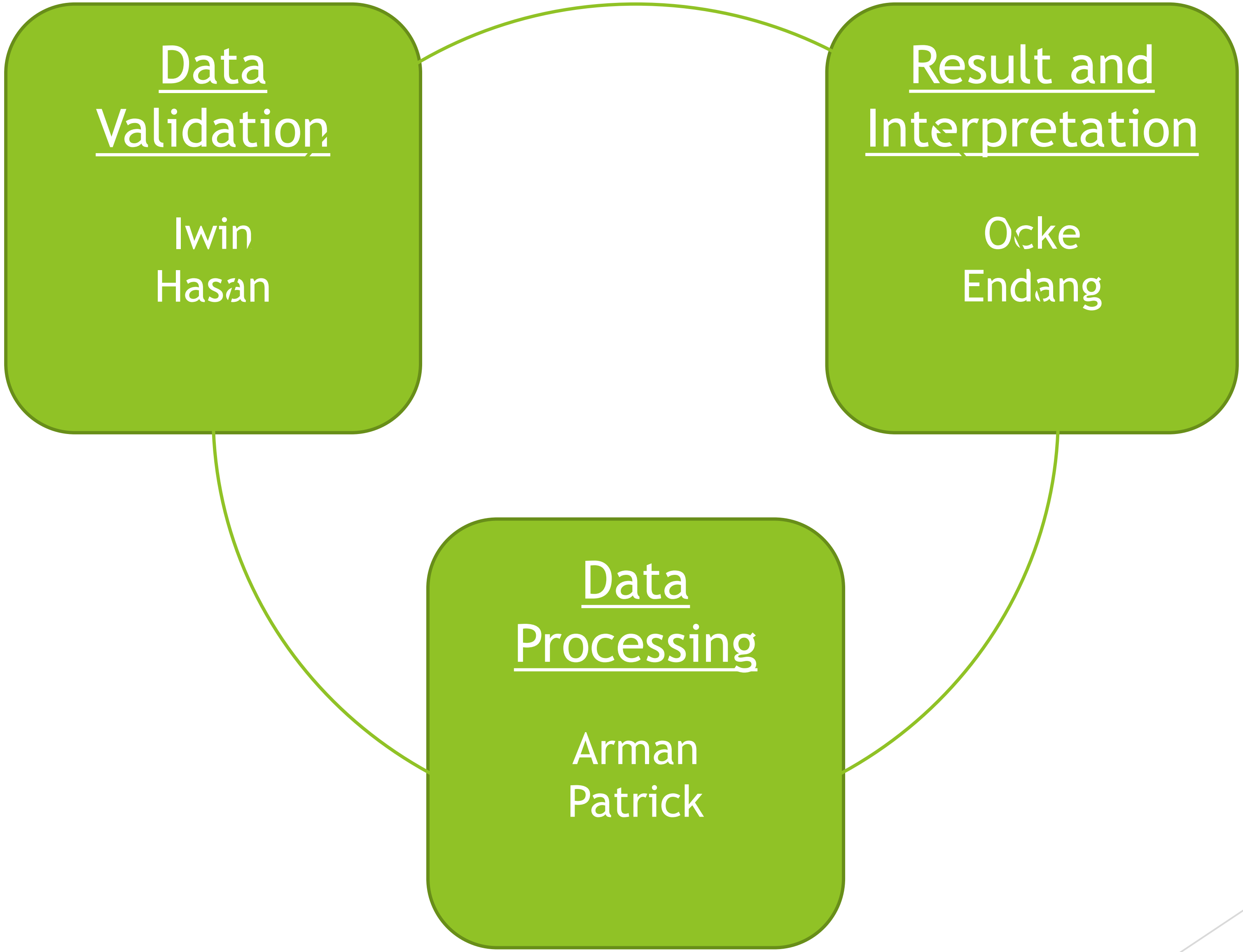
Claim Data Analytics

Lapse Team

Team Member



Data Validation

Iwin
Hasan

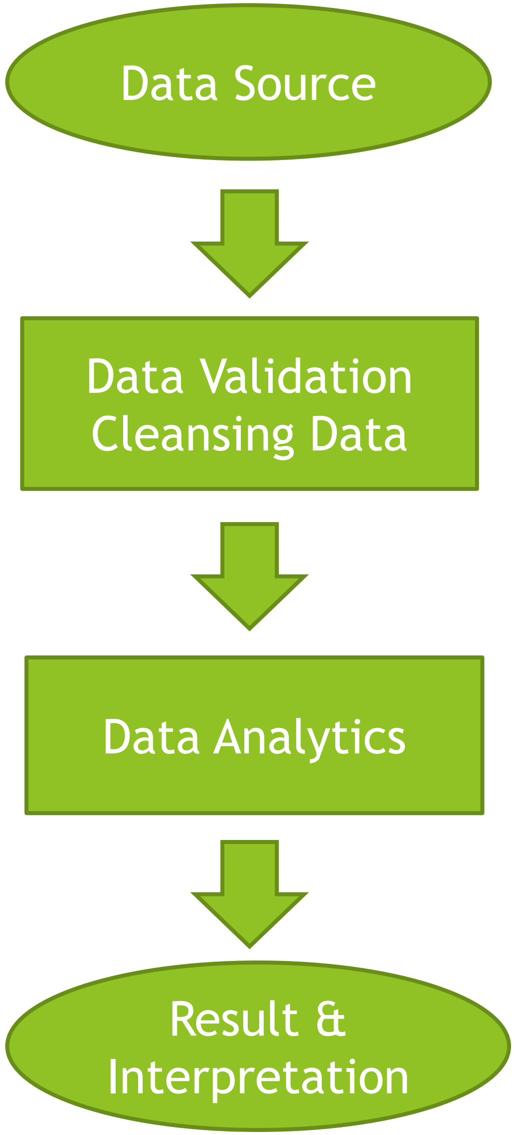
Result and Interpretation

Ocke
Endang

Data Processing

Arman
Patrick

Working Flow

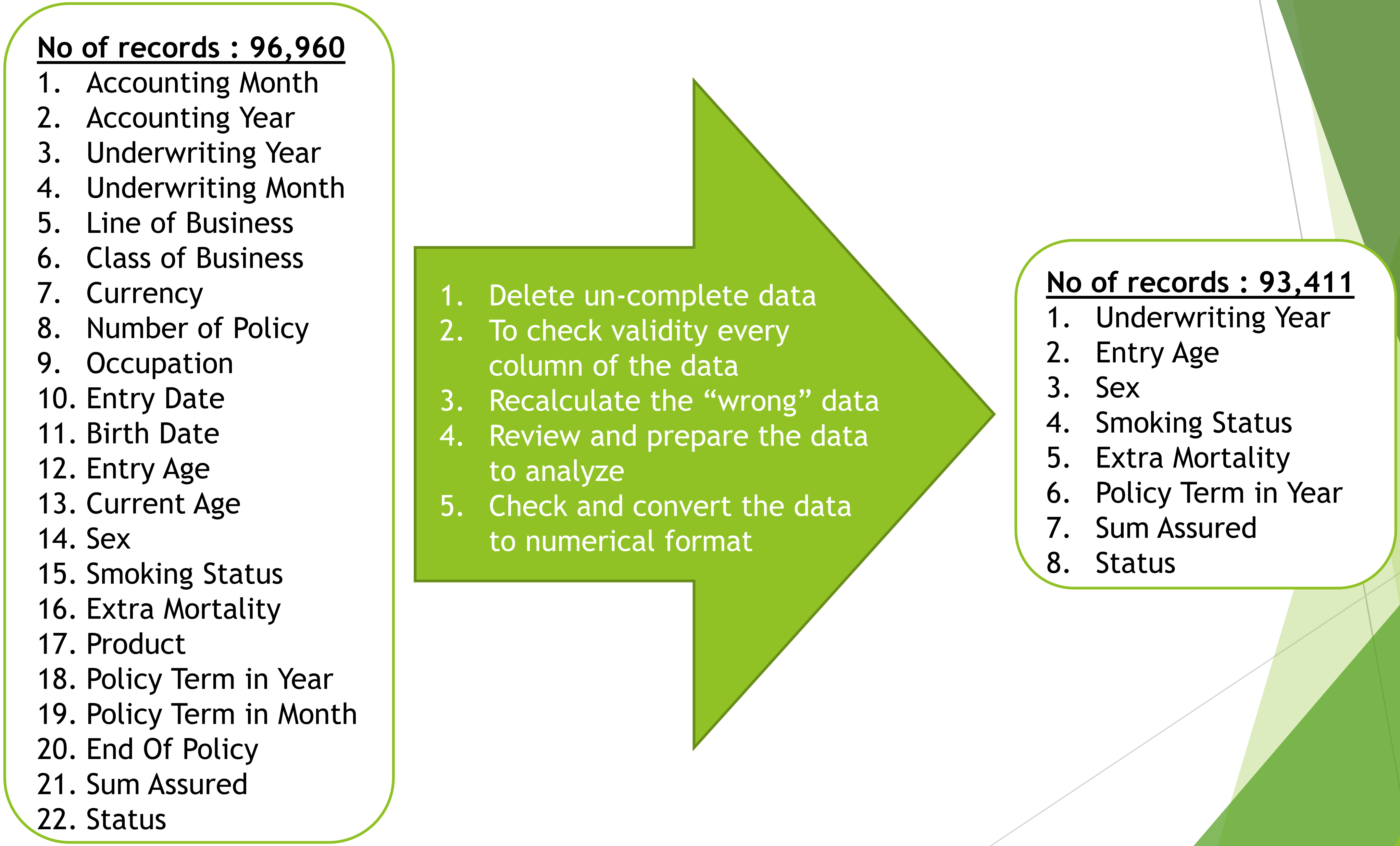

Data Source

Data Validation
Cleansing Data

Data Analytics

Result &
Interpretation

Data Validation



No of records : 96,960

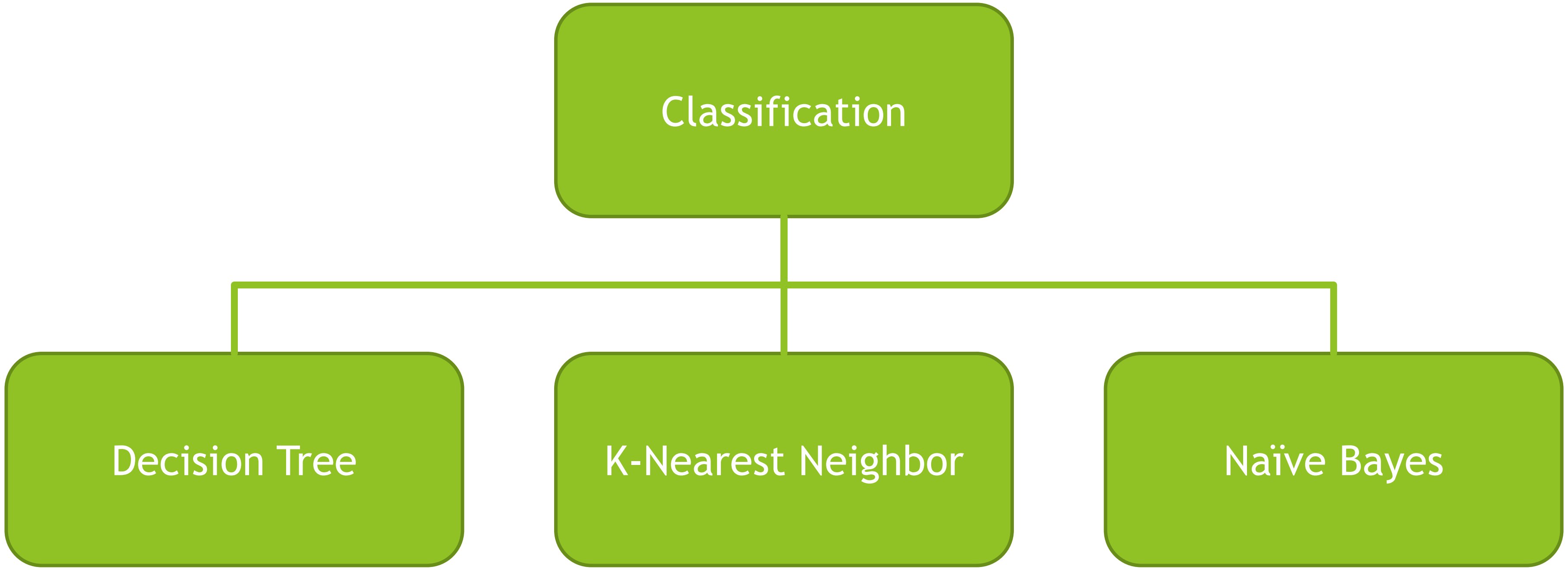
1. Accounting Month
2. Accounting Year
3. Underwriting Year
4. Underwriting Month
5. Line of Business
6. Class of Business
7. Currency
8. Number of Policy
9. Occupation
10. Entry Date
11. Birth Date
12. Entry Age
13. Current Age
14. Sex
15. Smoking Status
16. Extra Mortality
17. Product
18. Policy Term in Year
19. Policy Term in Month
20. End Of Policy
21. Sum Assured
22. Status

1. Delete un-complete data
2. To check validity every column of the data
3. Recalculate the “wrong” data
4. Review and prepare the data to analyze
5. Check and convert the data to numerical format

No of records : 93,411

1. Underwriting Year
2. Entry Age
3. Sex
4. Smoking Status
5. Extra Mortality
6. Policy Term in Year
7. Sum Assured
8. Status

Data Processing - Methodology



Classification

Decision Tree

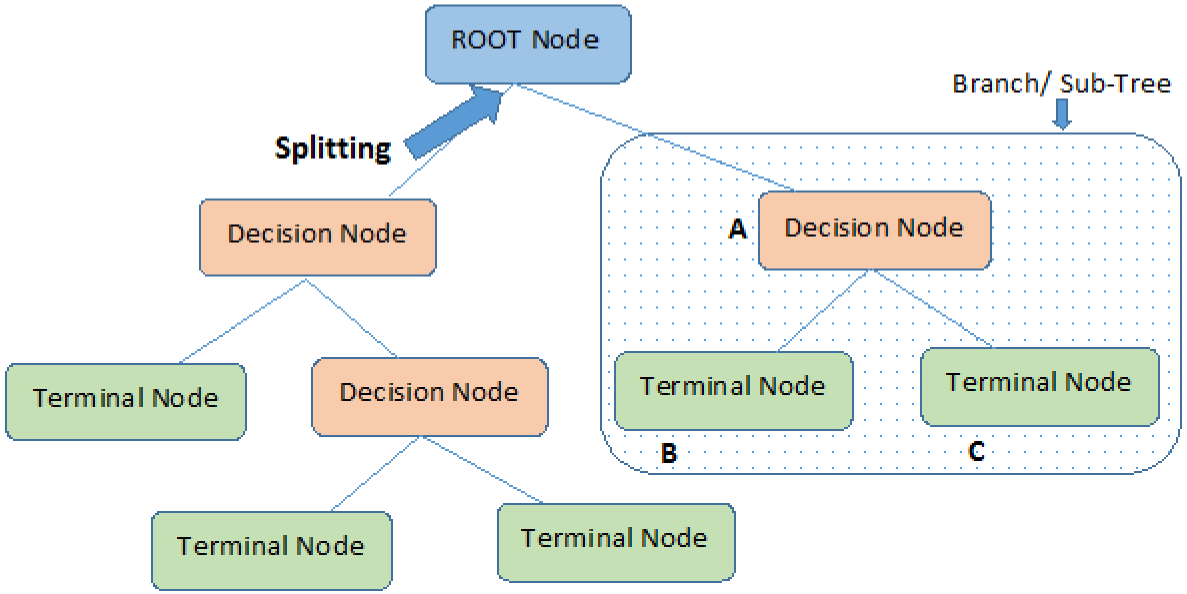
K-Nearest Neighbor

Naïve Bayes

Decision Tree

- **Decision trees** is one of the classification method. The basic idea behind decision trees is you make a set of questions that can partition your data set. You choose the question that provides the best split and again find the best questions for the partitions. You stop once all the points you are considering are of the same class.
- Then the task of classication is easy. You can simply grab a point, and chuck it down the tree. The questions will guide it to its appropriate class.

Decision Tree



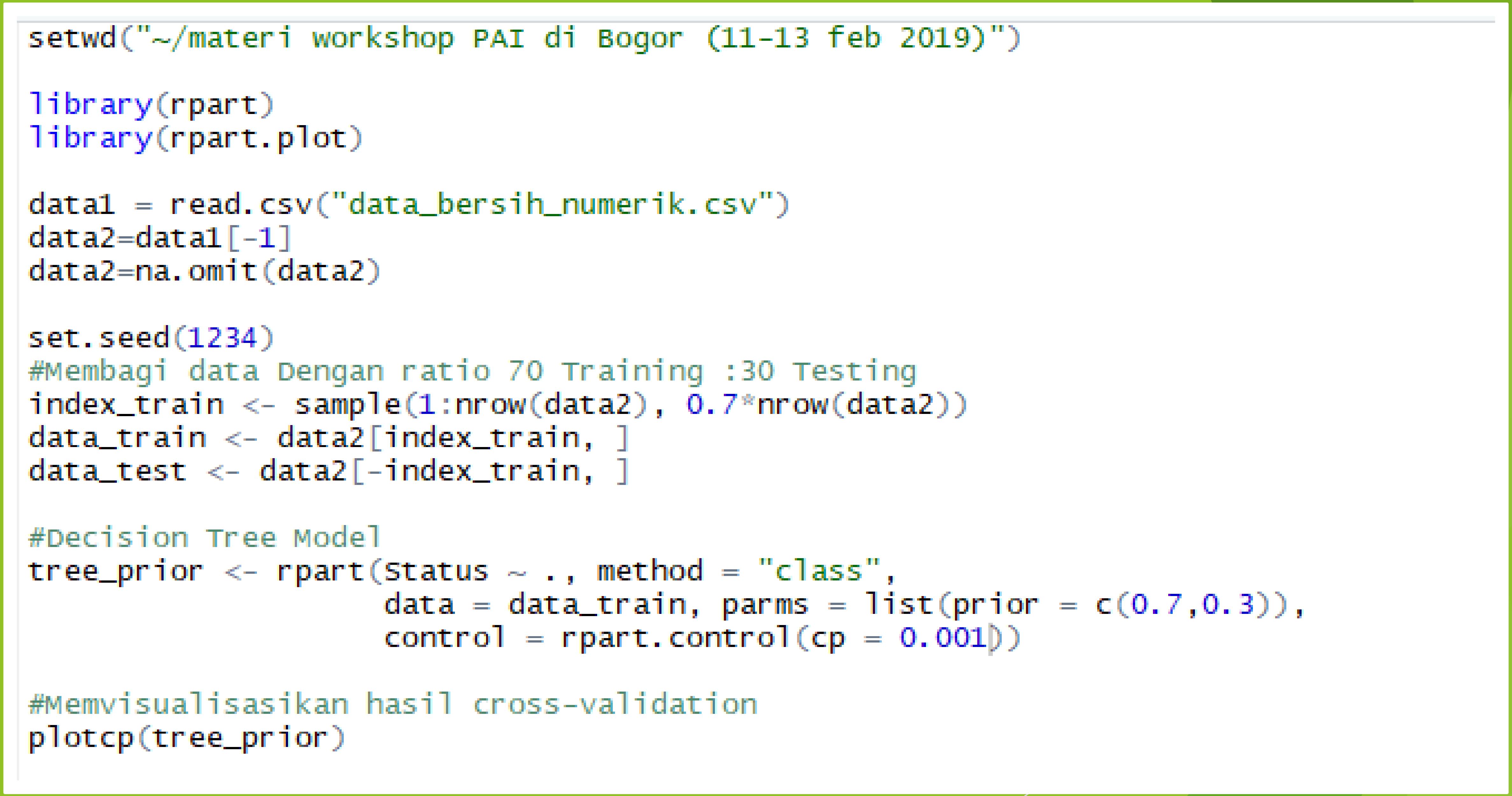
Note:- A is parent node of B and C.

Decision Tree

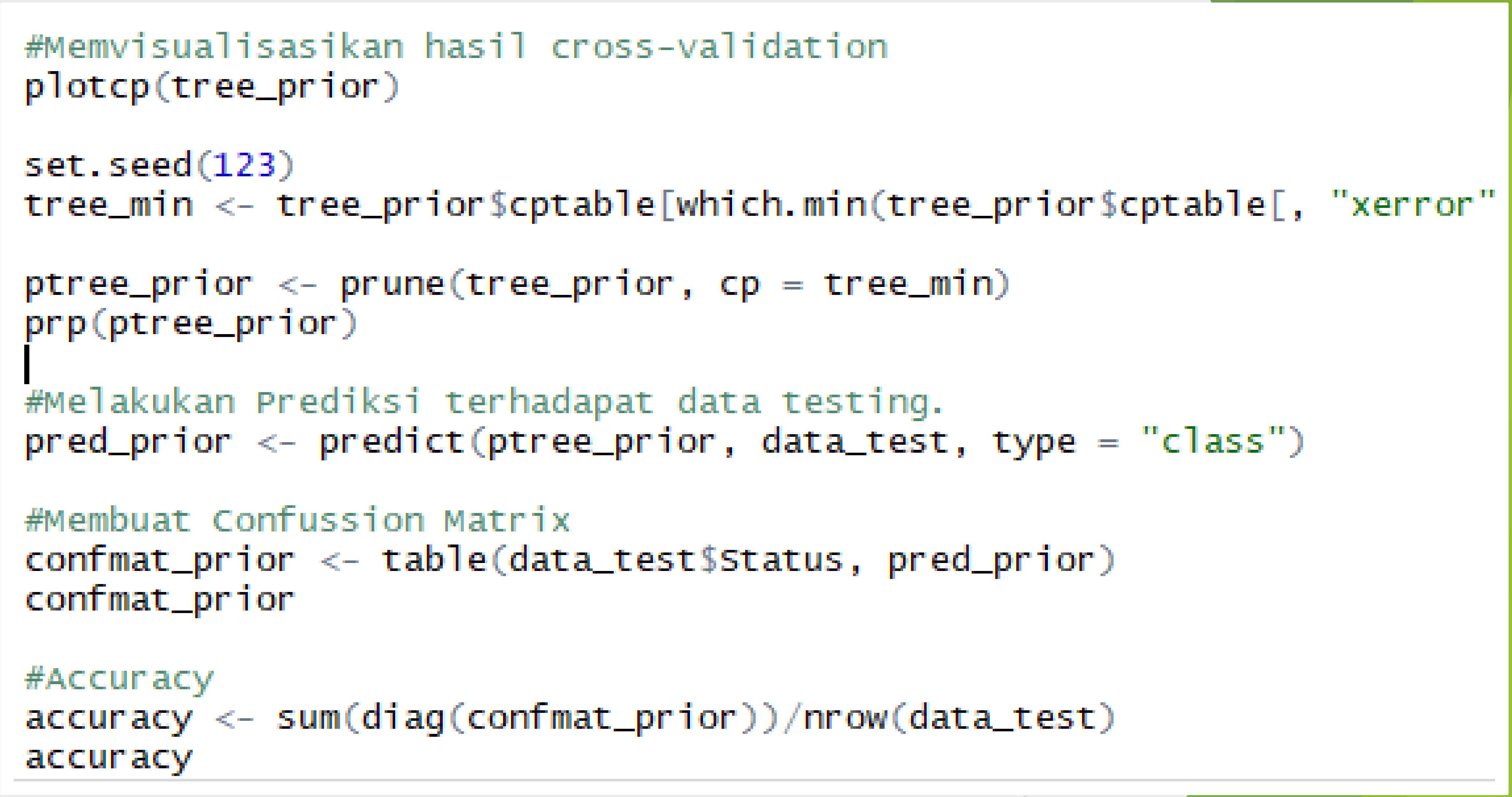
- Decision tree is a type of supervised learning algorithm that can be used in both regression and classification problems. It works for both categorical and continuous input and output variables.

Before using decision trees, consider it’s plus and minus

- The major advantage of using decision trees is that they are intuitively very easy to explain. They can be displayed graphically.
- One of the negative point of decision trees is: a small change in the data can cause a large change in the final estimated tree.



```
setwd("~/materi workshop PAI di Bogor (11-13 feb 2019)")

library(rpart)
library(rpart.plot)

data1 = read.csv("data_bersih_numerik.csv")
data2=data1[-1]
data2=na.omit(data2)

set.seed(1234)
#Membagi data Dengan ratio 70 Training :30 Testing
index_train <- sample(1:nrow(data2), 0.7*nrow(data2))
data_train <- data2[index_train, ]
data_test <- data2[-index_train, ]

#Decision Tree Model
tree_prior <- rpart(Status ~ ., method = "class",
                    data = data_train, parms = list(prior = c(0.7,0.3)),
                    control = rpart.control(cp = 0.001))

#Memvisualisasikan hasil cross-validation
plotcp(tree_prior)
```

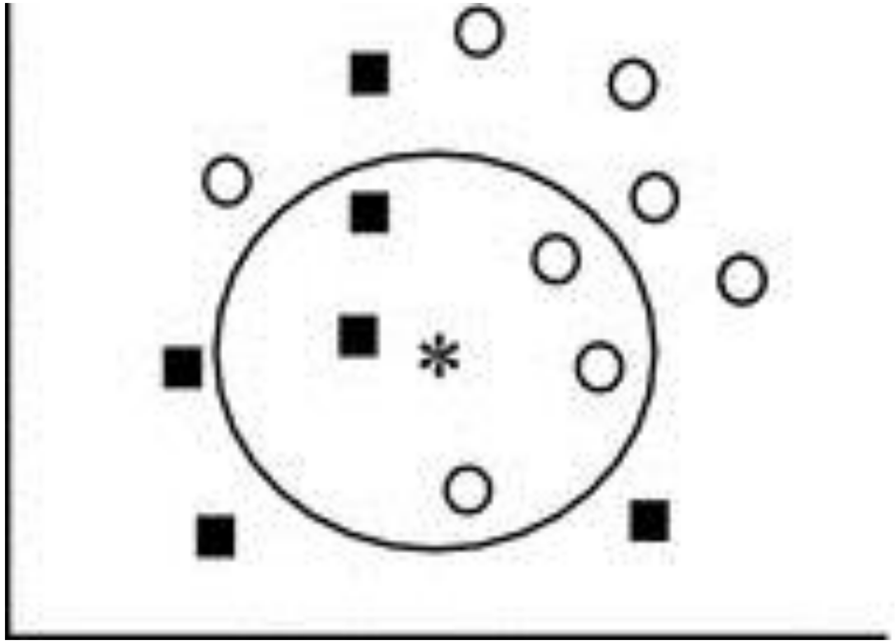


```r
#Memvisualisasikan hasil cross-validation
plotcp(tree_prior)

set.seed(123)
tree_min <- tree_prior$cptable[which.min(tree_prior$cptable[, "xerror"

ptree_prior <- prune(tree_prior, cp = tree_min)
prp(ptree_prior)

#Melakukan Prediksi terhadapat data testing.
pred_prior <- predict(ptree_prior, data_test, type = "class")

#Membuat Confussion Matrix
confmat_prior <- table(data_test$Status, pred_prior)
confmat_prior

#Accuracy
accuracy <- sum(diag(confmat_prior))/nrow(data_test)
accuracy
```

K-Nearest Neighbour

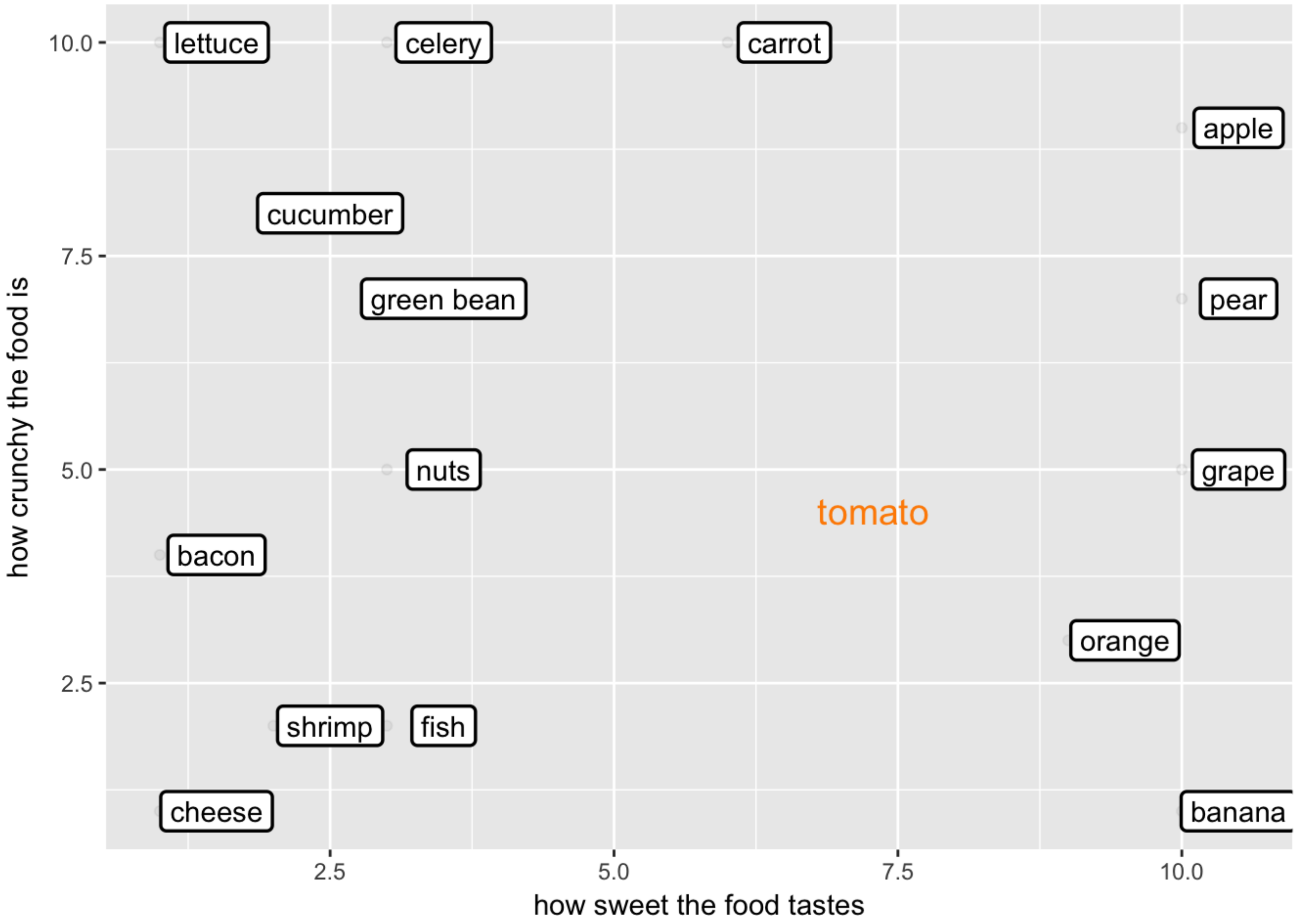
- The k-nearest neighbor algorithm gets it name from the fact that it uses information about an example's k-nearest neighbors to classify unlabeled examples.
- Upon choosing *k*, the algorithm requires a training dataset made up of examples that have been classified into several categories, as labeled by a nominal variable.
- Then, for each unlabeled record in the test dataset, k-NN identifies *k* records in the training data that are the "nearest" in similarity. The unlabeled test instance is assigned the class of the majority of the k-nearest neighbors.

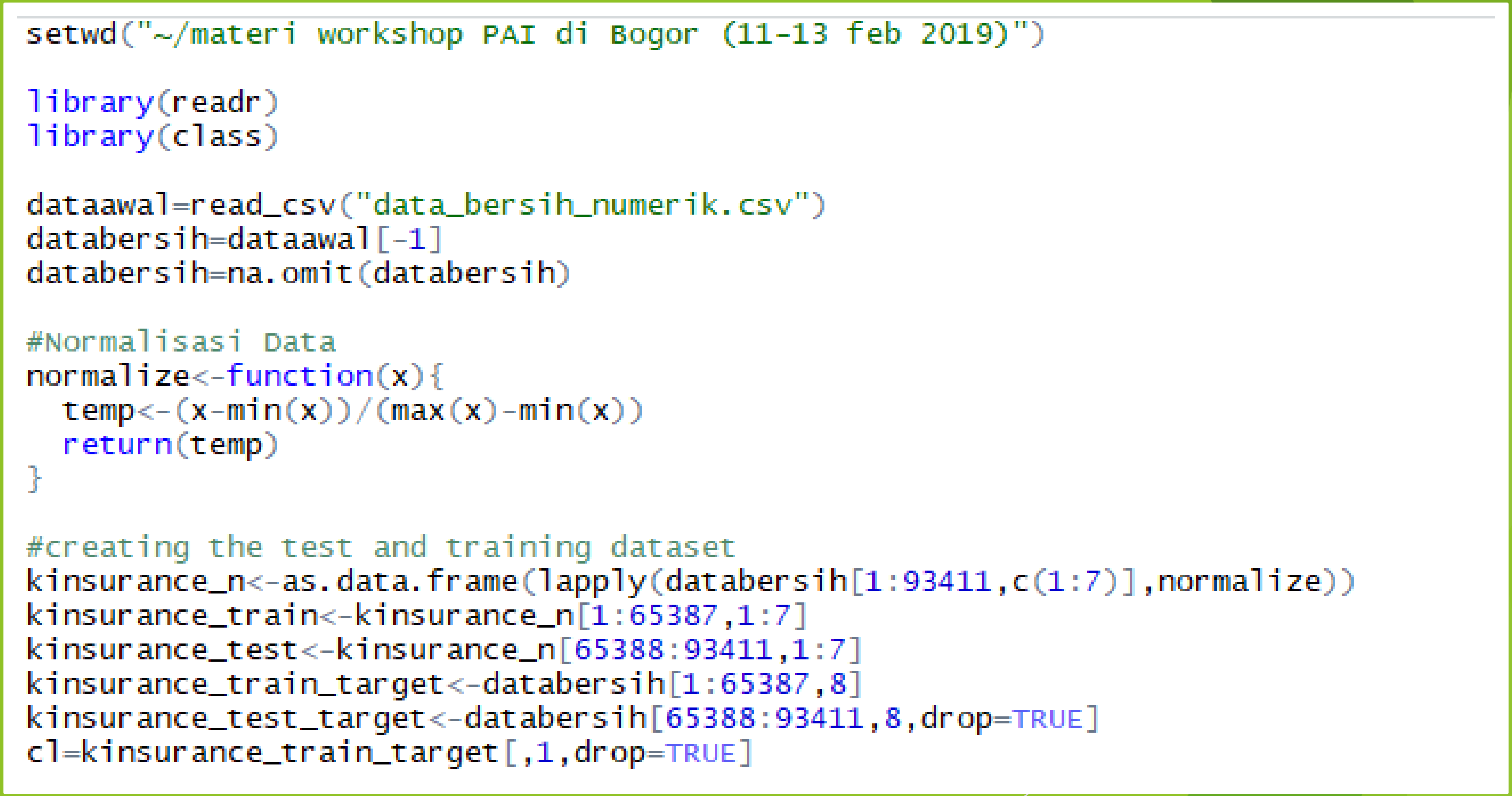
K-Nearest Neighbour



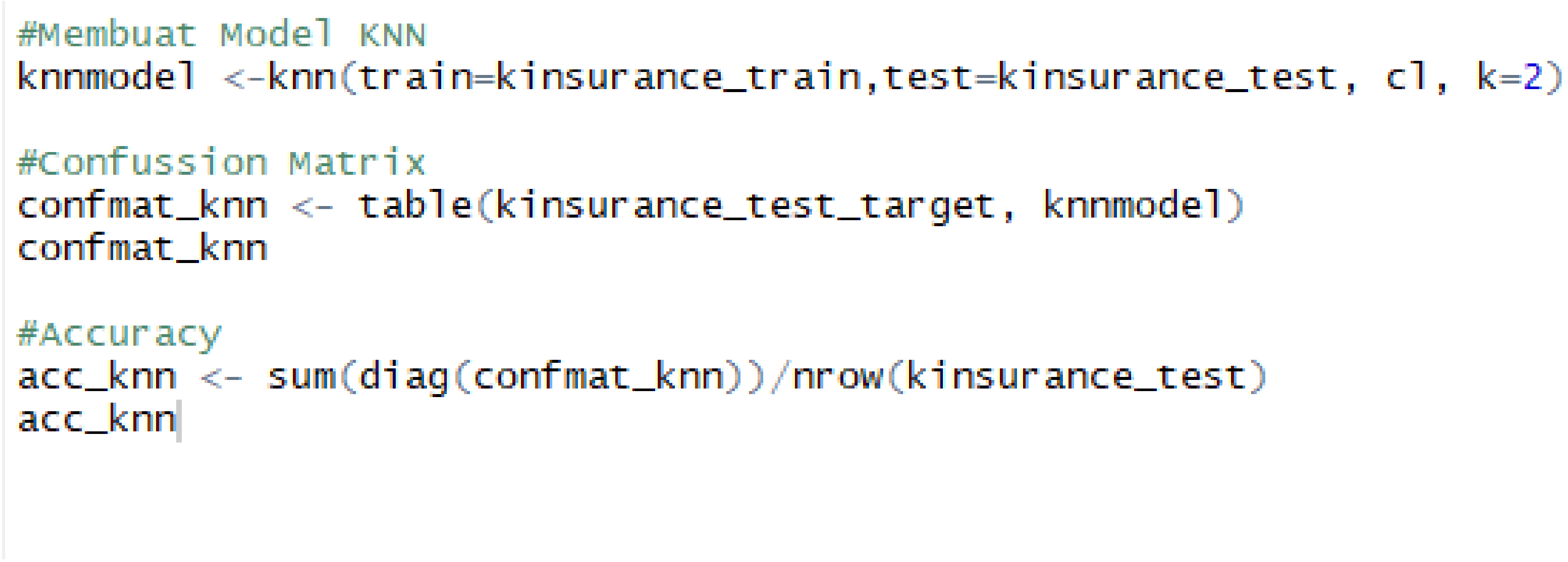
- Supposed we pick k=1, then the * in the following feature space will be assigned the square class, but if k=5, then the majority class of the five nearest point will be assigned to that point and our point will be classified as a round instead.

K-Nearest Neighbour





```r
setwd("~/materi workshop PAI di Bogor (11-13 feb 2019)")

library(readr)
library(class)

dataawal=read_csv("data_bersih_numerik.csv")
databersih=dataawal[-1]
databersih=na.omit(databersih)

#Normalisasi Data
normalize<-function(x){
  temp<-(x-min(x))/(max(x)-min(x))
  return(temp)
}

#creating the test and training dataset
kinsurance_n<-as.data.frame(lapply(databersih[1:93411,c(1:7)],normalize))
kinsurance_train<-kinsurance_n[1:65387,1:7]
kinsurance_test<-kinsurance_n[65388:93411,1:7]
kinsurance_train_target<-databersih[1:65387,8]
kinsurance_test_target<-databersih[65388:93411,8,drop=TRUE]
cl=kinsurance_train_target[,1,drop=TRUE]
```



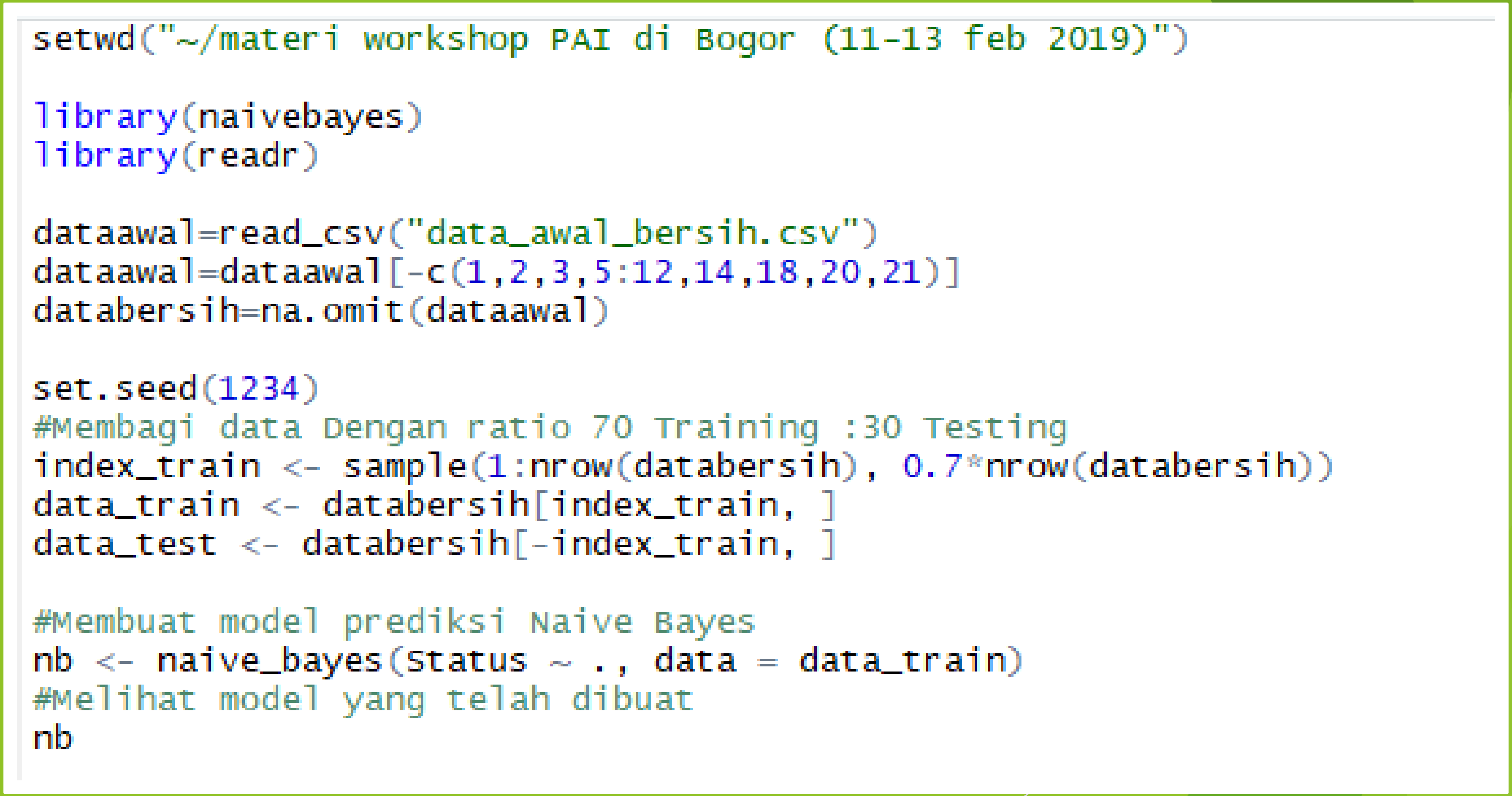
```
#Membuat Model KNN
knnmodel <-knn(train=kinsurance_train,test=kinsurance_test, cl, k=2)

#Confussion Matrix
confmat_knn <- table(kinsurance_test_target, knnmodel)
confmat_knn

#Accuracy
acc_knn <- sum(diag(confmat_knn))/nrow(kinsurance_test)
acc_knn
```

Naïve Bayes

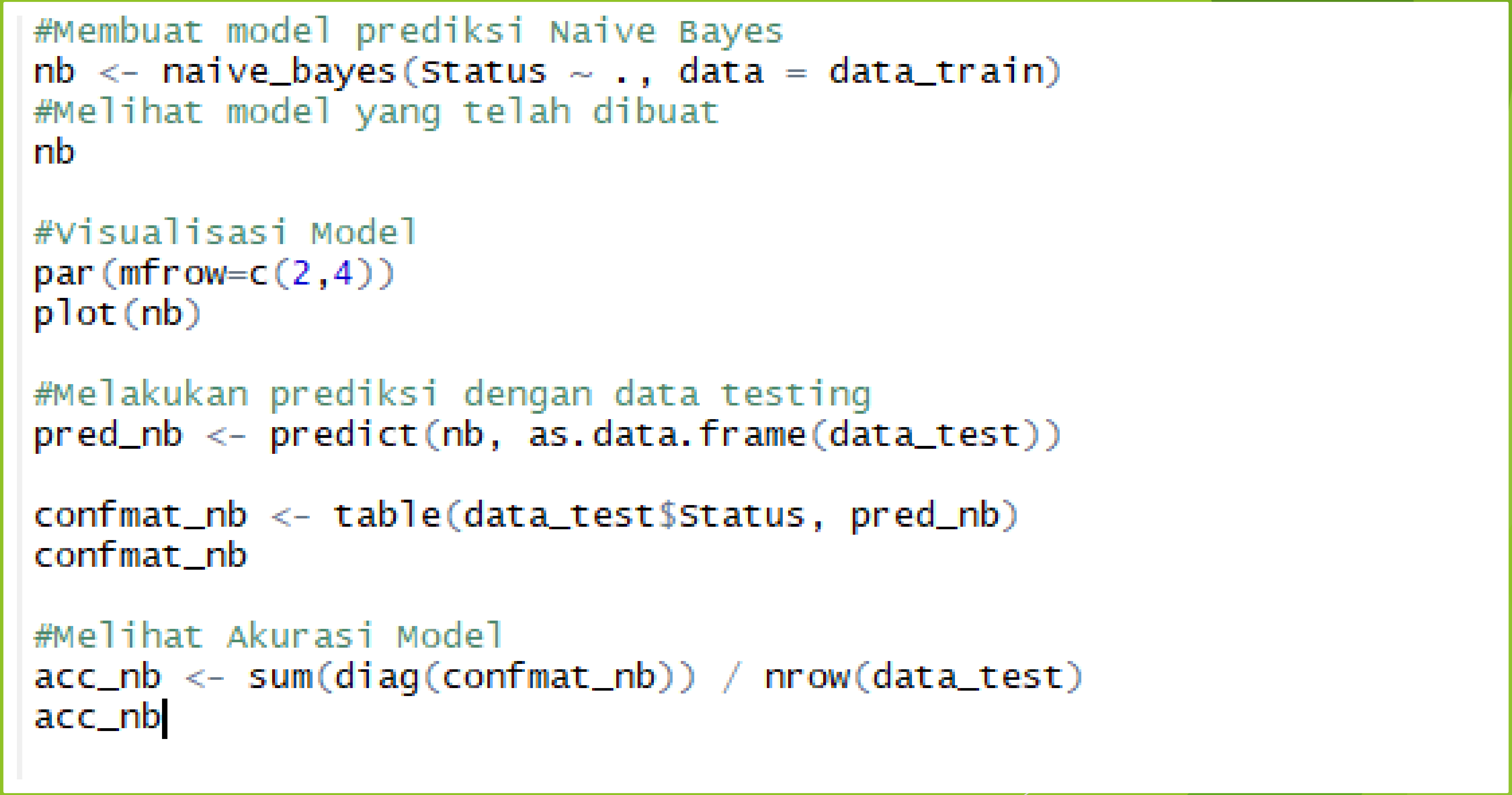
- Naive Bayes is a classifier which applies the well know Bayes theorem for conditional probability.
- In a classification problem, we have some predictors (features) and an outcome (target/class). Each observation has some values for the predictors and a class. From these predictors and associated classes we want to learn so that if the feature values are given, we can predict the class.
- In Naive Bayes, the algorithm evaluates a probability for each class, when the predictor values are given. And intuitively, we can go for the class, that has highest probability.

Naïve Bayes

- The reason that Naive Bayes algorithm is called Naive is not because it is simple or stupid. It is because the algorithm makes a very strong assumption about the data having features independent of each other while in reality, they may be dependent in some way.
- If this assumption of independence holds, Naive Bayes performs extremely well and often better than other models.
- Naive Bayes can also be used with continuous features but is more suited to categorical variables. If all the input features are categorical, Naive Bayes is recommended. However, in case of numeric features, it makes another strong assumption which is that the numerical variable is normally distributed.

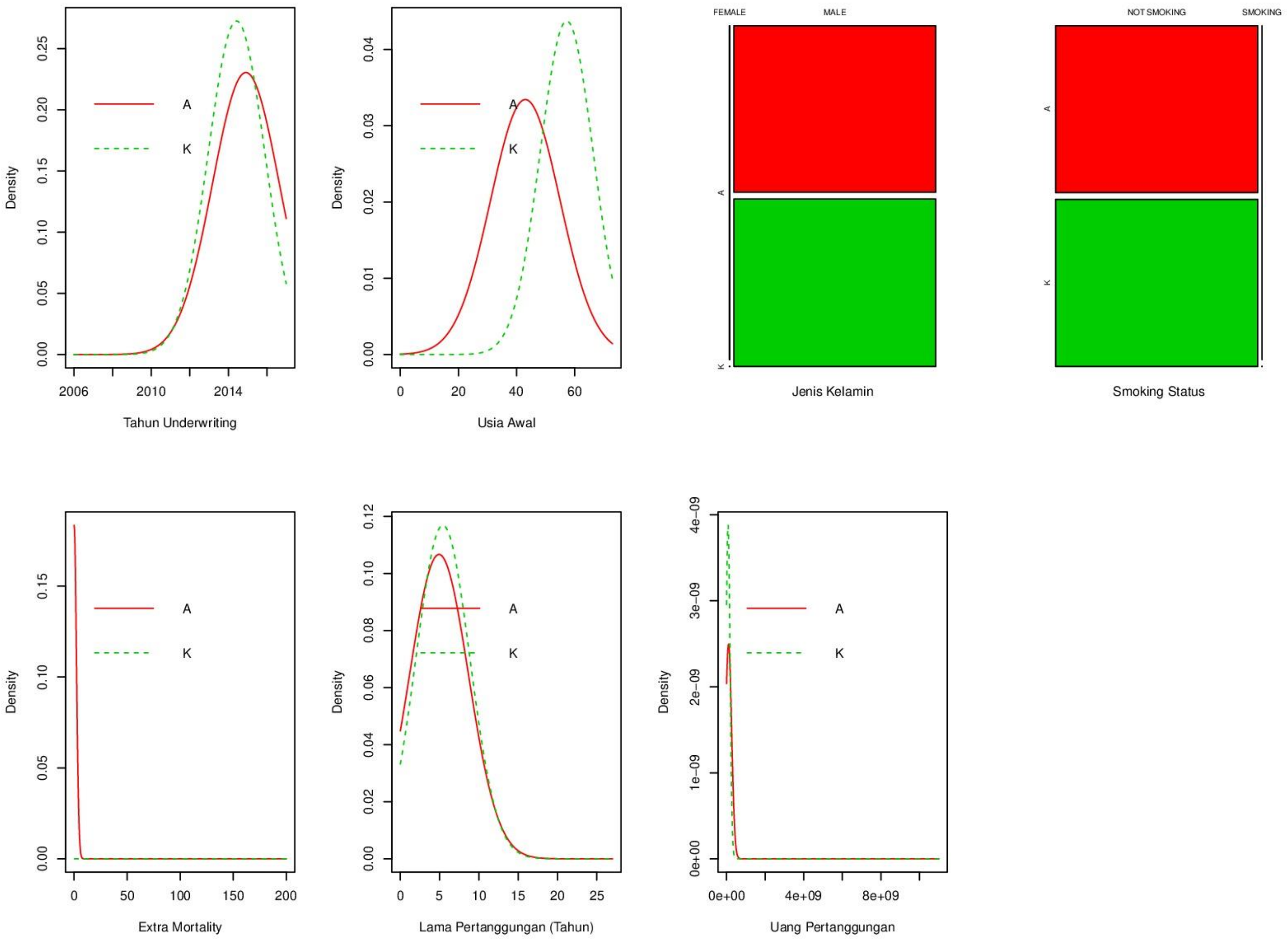


```r
setwd("~/materi workshop PAI di Bogor (11-13 feb 2019)")

library(naivebayes)
library(readr)

dataawal=read_csv("data_awal_bersih.csv")
dataawal=dataawal[-c(1,2,3,5:12,14,18,20,21)]
databersih=na.omit(dataawal)

set.seed(1234)
#Membagi data Dengan ratio 70 Training :30 Testing
index_train <- sample(1:nrow(databersih), 0.7*nrow(databersih))
data_train <- databersih[index_train, ]
data_test <- databersih[-index_train, ]

#Membuat model prediksi Naive Bayes
nb <- naive_bayes(Status ~ ., data = data_train)
#Melihat model yang telah dibuat
nb
```

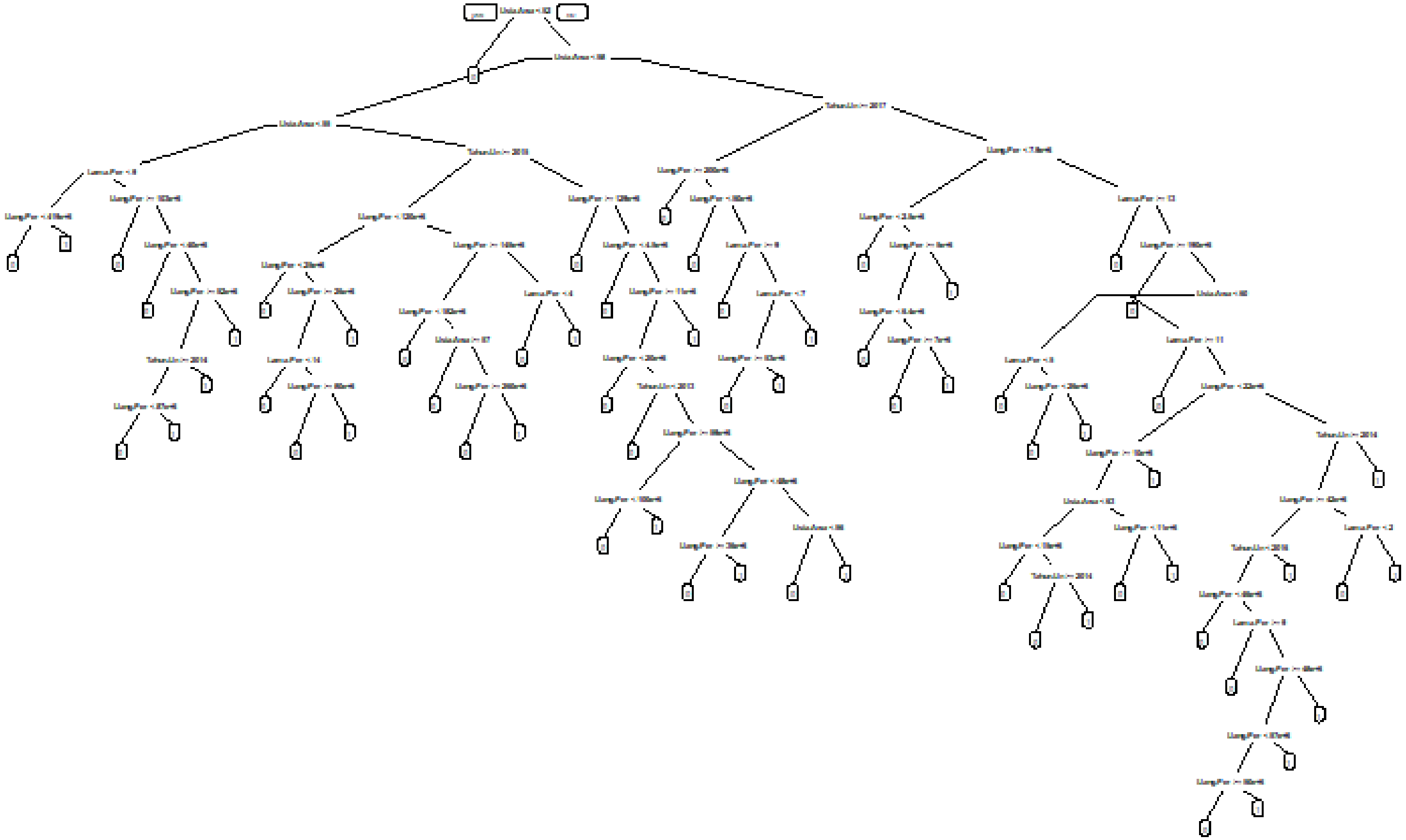


```
#Membuat model prediksi Naive Bayes
nb <- naive_bayes(Status ~ ., data = data_train)
#Melihat model yang telah dibuat
nb

#Visualisasi Model
par(mfrow=c(2,4))
plot(nb)

#Melakukan prediksi dengan data testing
pred_nb <- predict(nb, as.data.frame(data_test))

confmat_nb <- table(data_test$Status, pred_nb)
confmat_nb

#Melihat Akurasi Model
acc_nb <- sum(diag(confmat_nb)) / nrow(data_test)
acc_nb
```

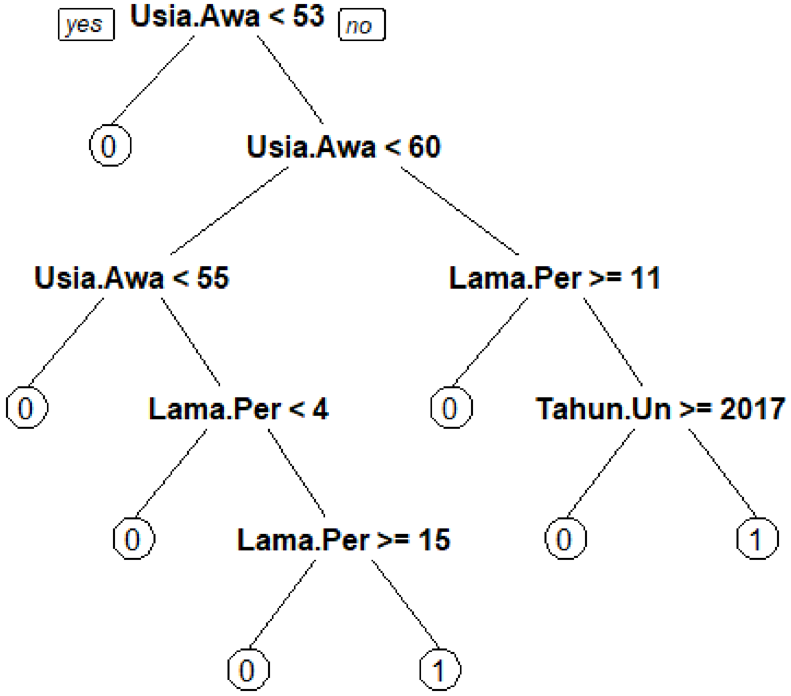
Output and Solution – Naïve Bayes

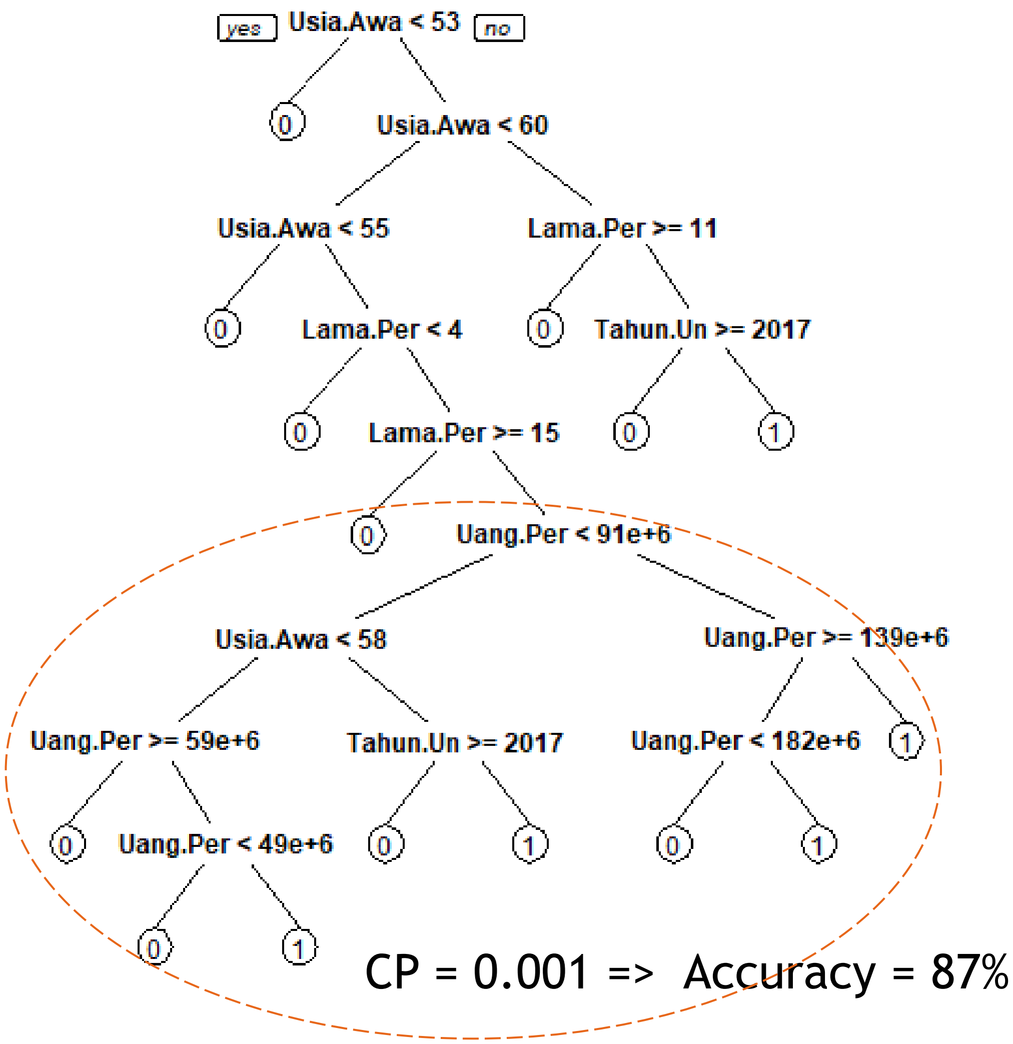


Output and Solution – Decision Tree



Output and Solution – Decision Tree





CP = 0.001 => Accuracy = 87%

Conclusion

Naïve Bayes Method

- There are several attributes that can be ignored. For example gender and smoking status. While there are seven attributes that we used. Three of them are underwriting years, initial age, duration of coverage.
- The accuracy is around 23.2% because the prior probabilities are not balanced. The probability of Claim is 0.02036 and probability of Non Claim is 0.9976.
- Another reason that accuracy is low that there are attributes of data continuously distributed which is approached with normal distribution, whereas the data is not normally distributed.

Conclusion

K-Nearest Neighbor

- Goal -> Find the value of k : is the number of the nearest point to determine the class: active, claims or terminate
- From data, level of accuracy is 98.20% when taken k = 2. This is better than k = 1
- When k = 1 the accuracy is high because the content of the target tends to be uniform. Number of K is more than A and T.

Conclusion

Decision Tree

- With the control parameter 0.001, we got the accuracy of the decision tree around 88.3%

Conclusion

- Tree Diagram and k-Nearest Neighbor models are more suitable and accurate to apply to the data. The two models can predict claims or not from new data in the future time.

Suggestion

- In the insurance industry there are any experiences data. So that, what we need to study is the attributes, like “job”. “Job” in this data have to be more specific.

References

- Prof. Andry Alamsyah dan team presentation material
- https://www.datacamp.com/community/tutorials/decision-trees-R
- https://www.datacamp.com/community/tutorials/machine-learning-in-r
- https://rpubs.com/riazakhan94/naive_bayes_classifier_e1071
- https://www.r-bloggers.com/understanding-naive-bayes-classifier-using-r/