

# EXP 16: Self-Organizing Maps (SOM) - Wine Quality Dataset

## AIM

To apply and evaluate the **Self-Organizing Map (SOM)** algorithm on the **Wine Quality dataset** to visualize patterns in high-dimensional data and predict wine quality scores.

## EXPERIMENTAL SETUP

- Dataset: Wine Quality (Red Wine) from UCI Repository
- Libraries used: `ggplot2`, `kohonen`, `caret`
- Objective: Visualize data clusters and evaluate prediction accuracy using **Mean Squared Error (MSE)**.

## LIBRARIES REQUIRED

```
library(ggplot2)    # For visualization
library(kohonen)    # For Self-Organizing Maps
library(caret)      # For data partitioning
```

```
## Loading required package: lattice
```

```
# Load the wine quality dataset
```

```
wine_data <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv")
```

```
# View structure and summary
```

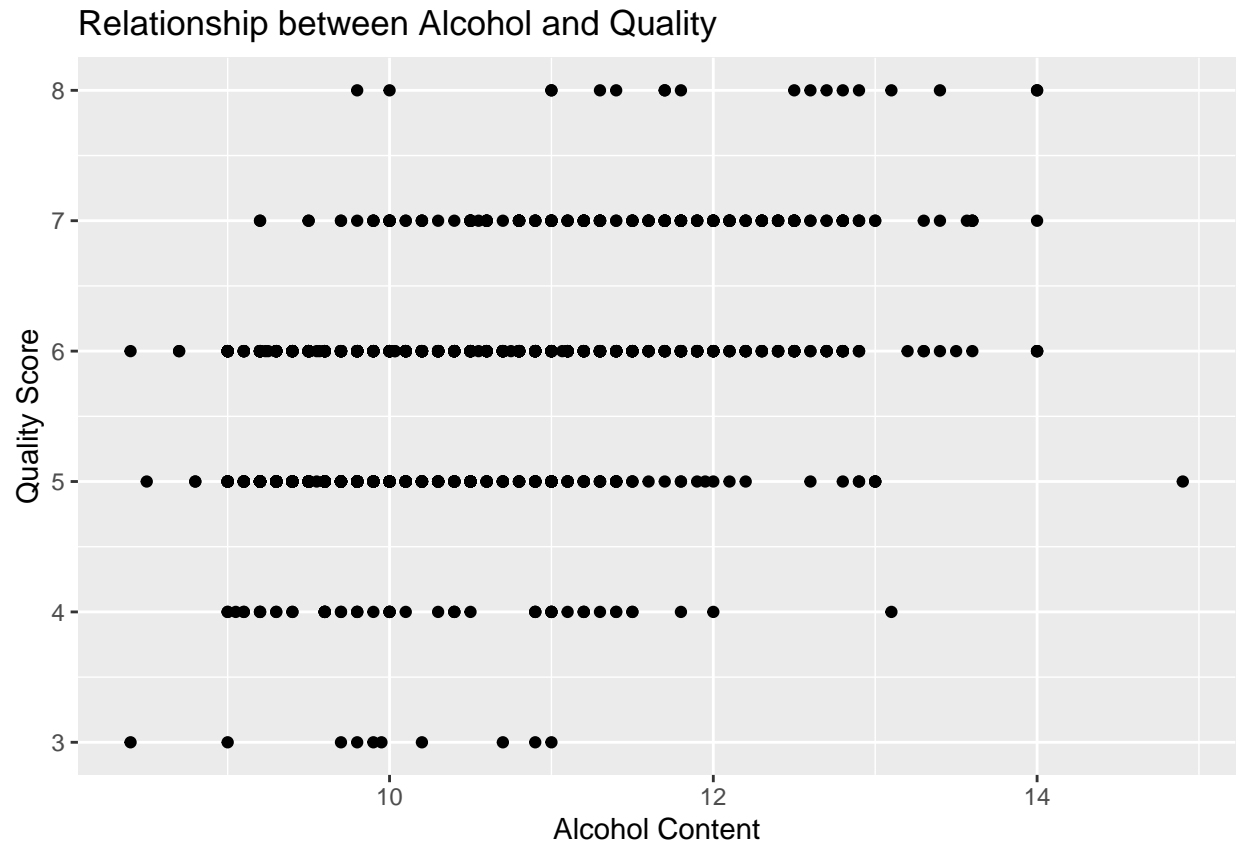
```
str(wine_data)
```

```
## 'data.frame':    1599 obs. of  12 variables:
## $ fixed.acidity      : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
## $ volatile.acidity   : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
## $ citric.acid        : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
## $ residual.sugar     : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
## $ chlorides          : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
## $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
## $ total.sulfur.dioxide : num  34 67 54 60 34 40 59 21 18 102 ...
## $ density            : num  0.998 0.997 0.997 0.998 0.998 ...
## $ pH                 : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
## $ sulphates          : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
## $ alcohol            : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
## $ quality             : int   5 5 5 6 5 5 5 7 7 5 ...
```

```
summary(wine_data)
```

```
## fixed.acidity  volatile.acidity  citric.acid  residual.sugar
## Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
## 1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
## Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
## Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
## 3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
## Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
## chlorides      free.sulfur.dioxide total.sulfur.dioxide density
## Min.   :0.01200   Min.   : 1.00      Min.   : 6.00      Min.   :0.9901
## 1st Qu.:0.07000   1st Qu.: 7.00      1st Qu.: 22.00     1st Qu.:0.9956
## Median :0.07900   Median :14.00      Median : 38.00     Median :0.9968
## Mean   :0.08747   Mean   :15.87      Mean   : 46.47     Mean   :0.9967
## 3rd Qu.:0.09000   3rd Qu.:21.00      3rd Qu.: 62.00     3rd Qu.:0.9978
## Max.   :0.61100   Max.   :72.00      Max.   :289.00     Max.   :1.0037
## pH             sulphates          alcohol          quality
## Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
## 1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
## Median :3.310   Median :0.6200   Median :10.20   Median :6.000
## Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
## 3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
## Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
```

```
# Visualize alcohol content vs. quality
ggplot(wine_data, aes(x = alcohol, y = quality)) +
  geom_point() +
  labs(title = "Relationship between Alcohol and Quality",
       x = "Alcohol Content",
       y = "Quality Score")
```



```
# Remove target variable before training SOM
wine_data_processed <- wine_data[, -ncol(wine_data)]

# Normalize the feature variables
wine_data_processed <- scale(wine_data_processed)

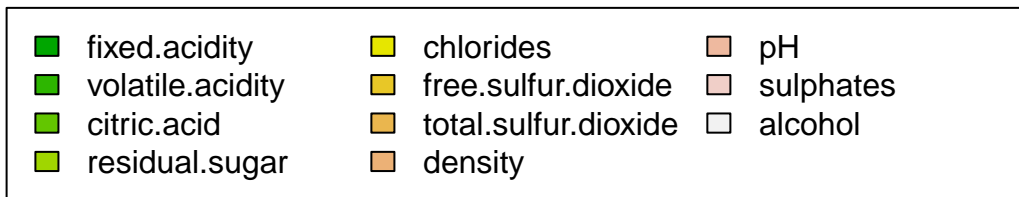
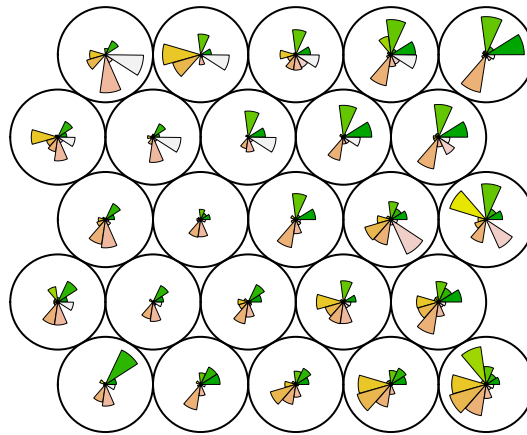
set.seed(123) # For reproducibility
train_index <- createDataPartition(wine_data$quality, p = 0.8, list = FALSE)

train_data <- wine_data_processed[train_index, ]
test_data <- wine_data_processed[-train_index, ]

# Train SOM model
som_model <- som(train_data, grid = somgrid(5, 5, "hexagonal"))

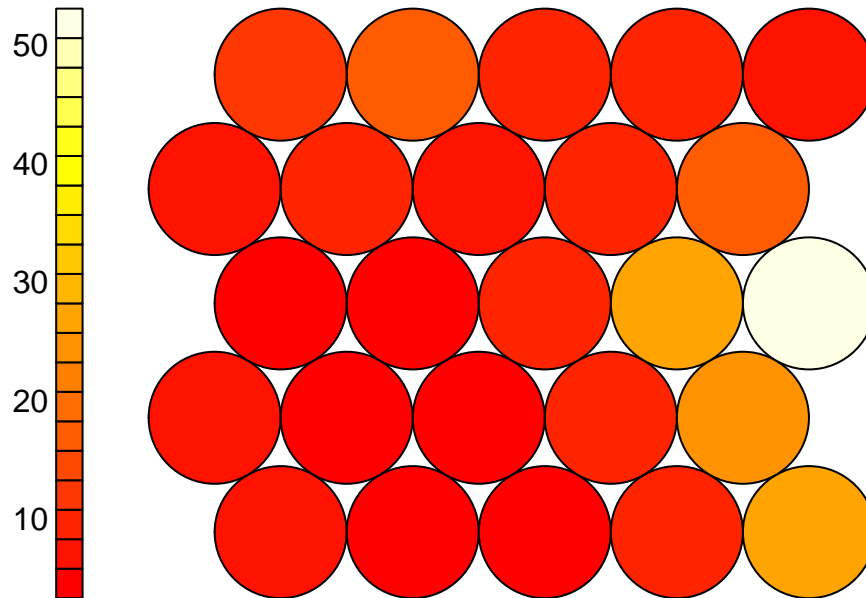
# Plot codebook vectors
plot(som_model, type = "codes", main = "SOM Codes")
```

## SOM Codes



```
# Plot neighbor distances  
plot(som_model, type = "dist.neighbours", main = "Distance to Neighbouring Units")
```

## Distance to Neighbouring Units



```
# Predict cluster units for test data
predictions <- predict(som_model, test_data)

# Compute Mean Squared Error
mse <- mean((predictions$unit.classif - wine_data$quality[-train_index])^2)
print(paste("Mean Squared Error:", mse))
```

```
## [1] "Mean Squared Error: 90.0786163522013"
```

## CONCLUSION

The SOM model successfully visualized complex relationships in the wine dataset. The Mean Squared Error indicates how closely the SOM units aligned with actual wine quality scores, helping assess its effectiveness in unsupervised learning tasks.