K-Nearest Neighbors Algorithm

Using Social Network Ad Data

Author : Rose Ellison

The Intuition

K-Nearest Neigbors(KNN) is a type of supervised machine learning algorithm which can be used for both classification and regression problems. It is typically used for predictive problems.

Step One: Split the data into test and training sets

Step Two : Choose the value K, which can be any integer.

Step Three : For each point in the test data do the following

- **3A :** Calculate the distance beteen test data and each row of training data. We will use the Euclidean distance method.

$$d(p, q) = \sqrt{\sum (q_i - p_i)^2}$$

- **3B :** Sort the distance values in ascending order
- **3C :** Choose the top K rows from the sorted array
- **3D :** Assign a class to the test point based on most frequent class of these rows.

Exploration

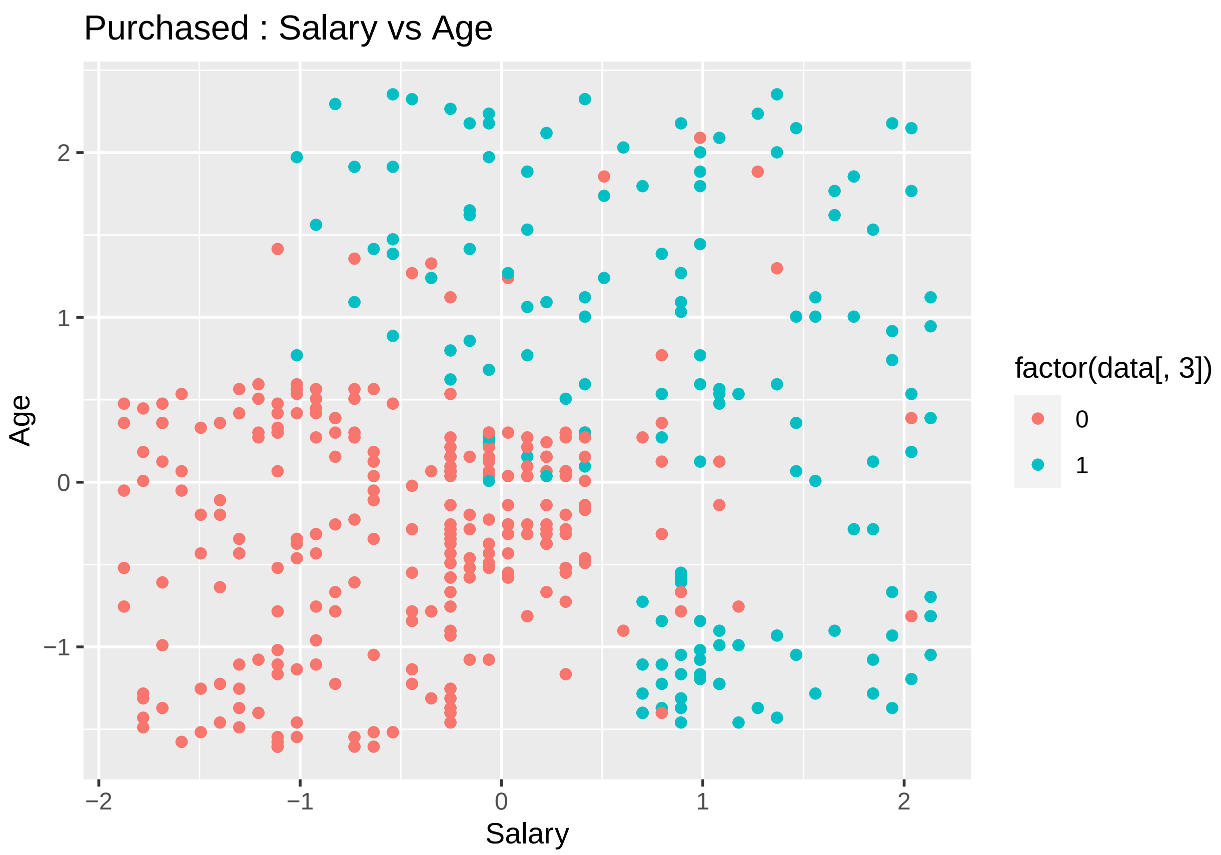
Exploration process includes preprocessing and visualizing plots. This algorithm will be implemented on social network ads data. There are three columns; purchased, salary, and age. We will be trying to predict if future customers will purchase a social network ad.

Preprocessing

```r
# Read the data
data = read.csv('../../../_resources/data/Social_Network_Ads.csv')
data <- data[, 3:5]
# Feature Scaling
data[-3] = scale(data[-3])
```

Plot

```
ggplot(data,aes(x = data[, 1],
                y = data[, 2],
                color = factor(data[ , 3]))) +
  geom_point() +
  labs(x = "Salary",
       y = "Age") +
  ggtitle("Purchased : Salary vs Age")
```



From the plot, we can see this data is linearly seperable. The KNN algorithm classifier should give us the line of best fit that best seperates this data.

Implementation

Step One :

Split the data into test and training sets.

```
# Splitting test and training sets
split <- sample.split(data[ , 3], SplitRatio = 0.8)
training <- subset(data, split == TRUE)
test <- subset(data, split == FALSE)
```

Step Two :

Choose the value K, which can be any integer.

```r
k <- 5
```

Step Three :

For each point in the test data do the following, 1.) Calculate the distance beteen test data and each row of training data. We will use the Euclidean distance method. 2.) Sort the distance values in ascending order. 3.) Choose the top K rows from the sorted array. 4.) Assign a class to the test point based on most frequent class of these rows.

3A.)

```r
# Euc distance per point
get_euc_dist <- function(p, q)
{
  # Euclidean Distance calculation
  return(sqrt(sum(p - q) ^ 2))
}
```

4A.)

```r
knn <- function(data, k)
{

  for (i in 1: nrow(data))
  {
    # Loop through all points
    for (j in 1:nrow(data))
    {
      # Save distances in list
      data$Distance[i] <- get_euc_dist(data[j, 1], data[j, 2])

    }

    neighbors <- data %>%
                  arrange(Distance) %>%
                  head(k)

    zero_count <- sum(neighbors$Purchased == 0)
    one_count <- sum(neighbors$Purchased == 1)


    if(zero_count > one_count)
    {
      data$y_pred[i] <- 0
    } else
    {
      data$y_pred[i] <- 1
    }
  }
  return(data$y_pred)
}

training$y_pred <- knn(training, k)
```

Results

```
training %>%
  select(y_pred, Purchased)
```

```
##      y_pred Purchased
## 1         0         0
## 2         0         0
## 3         0         0
## 5         0         0
## 7         0         0
## 8         0         1
## 10        0         0
## 12        0         0
## 13        1         0
## 14        1         0
## 15        1         0
## 18        1         1
## 19        1         1
## 20        1         1
## 22        1         1
## 23        0         1
## 25        0         1
## 26        0         1
## 29        0         0
## 30        0         0
## 31        0         0
## 32        0         1
## 33        0         0
## 38        0         0
## 39        0         0
## 40        0         0
## 41        0         0
## 42        0         0
## 43        0         0
## 47        0         0
## 48        0         0
## 49        0         1
## 50        0         0
## 51        0         0
## 52        0         0
## 54        0         0
## 55        0         0
## 56        0         0
## 57        0         0
## 58         0          0
## 59         0          0
## 61         0          0
## 63         0          0
## 64         0          1
## 65         0          0
## 66         0          0
## 67         0          0
## 68         0          0
## 69         0          0
## 70         0          0
## 71         0          0
## 73         0          0
## 74         0          0
## 75         0          0
## 76         0          1
## 77         0          0
## 80         0          0
## 81         0          0
## 84         0          0
## 85         0          0
## 88         0          0
## 89         0          0
## 90         0          0
## 91         0          0
## 92         0          0
## 94         0          0
## 95         0          0
## 96         0          0
## 97         0          0
## 98         0          1
## 99         0          0
## 101        0          0
## 102        0          0
## 103        0          0
## 104        0          1
## 106        0          0
## 107        0          0
## 108        0          0
## 110        0          0
## 111        0          0
## 112        0          0
## 113        0          0
## 114        0          0
## 115        0          0
## 116        0          0
## 118        0          0
## 119        0          0
## 120        0          0
## 121        0          0
## 122        0          0
## 124        0          0
## 125        0          0
## 126        0          0
## 127        0          0
## 128        0          0
## 130        0          0
## 132        0          0
## 133        0          0
## 134        0          0
## 136        0          0
## 137        0          0
## 138        0          1
## 139        0          0
## 141        0          0
## 142        0          0
## 143        0          0
## 146        0          0
## 147        0          1
## 148        0          0
## 149        0          0
## 150        0          0
## 151        0          0
## 153        0          0
## 154        0          0
## 155        0          0
## 156        0          0
## 157        0          0
## 158        0          0
## 159        0          0
## 161        0          1
## 162        0          0
## 164        0          0
## 165        0          0
## 166        0          0
## 167        0          0
## 168        0          0
## 169        0          1
## 171        0          0
## 172        0          0
## 173        0          0
## 174        0          0
## 175        0          0
## 176        0          0
## 177        0          0
## 178        0          0
## 180        0          0
## 181        0          0
## 182        0          0
## 183        0          1
## 184        0          0
## 185        0          0
## 186        0          0
## 188        0          0
## 189        0          0
## 190        0          0
## 193        0          0
## 194        0          0
## 195        0          0
## 196        0          0
## 197        0          0
## 198        0          0
## 199        0          0
## 200        0          0
## 201        1          0
## 202        1          0
## 203        1          1
## 204        1          0
## 205        1          1
## 207        0          1
## 208        0          0
## 209        0          1
## 210        0          0
## 211        0          1
## 213        0          0
## 214        0          0
## 215        0          0
## 216        0          1
## 217        0          0
## 218        1          0
## 219        1          0
## 220        1          1
## 221        1          0
## 222        0          1
## 223        0          1
## 224        0          1
## 225        0          0
## 226        1          0
## 229        1          0
## 230        1          1
## 231        0          1
## 232        0          0
## 234        0          1
## 235        0          0
## 236        0          1
## 237        1          0
## 238        1          0
## 239        1          0
## 240        1          1
## 241        1          1
## 243        0          1
## 245        1          0
## 246        1          1
## 247        1          0
## 248        1          1
## 249        1          0
## 250        1          1
## 253        0          1
## 254        0          1
## 255        0          0
## 256        0          1
## 257        1          0
## 258        1          0
## 260        1          1
## 261        1          0
## 262        1          1
## 263        0          1
## 264        0          0
## 265        0          1
## 266        0          1
## 267        0          0
## 268        1          0
## 270        1          0
## 271        1          0
## 272        1          1
## 273        1          1
## 275        1          1
## 276        1          1
## 277        1          0
## 278        1          1
## 279        1          1
## 280        1          1
## 281        1          1
## 283        0          1
## 285        1          0
## 286        1          1
## 287        1          0
## 288        1          1
## 289        1          0
## 290        1          1
## 291        0          1
## 292        0          1
## 293        0          1
## 294        0          0
## 295        1          0
## 296        1          0
## 297        1          1
## 298        1          1
## 299        1          0
## 300        1          1
## 301        1          1
## 302        1          1
## 304        0          1
## 305        1          0
## 306        1          0
## 308        1          1
## 309        0          1
## 310        1          0
## 312        1          1
## 313        1          0
## 314        1          1
## 315        1          0
## 316        1          1
## 317        0          1
## 318        1          0
## 319        1          1
## 320          1               0
## 321          0               1
## 323          0               0
## 324          0               1
## 325          0               1
## 326          0               0
## 327          0               0
## 328          1               0
## 329          0               1
## 330          0               1
## 331          0               0
## 334          0               0
## 335          0               1
## 336          0               0
## 337          0               1
## 338          0               0
## 339          0               0
## 340          0               1
## 342          1               0
## 343          1               0
## 345          0               1
## 346          1               0
## 347          0               1
## 348          0               1
## 349          0               0
## 350          1               0
## 351          1               1
## 352          1               0
## 353          1               1
## 354          1               0
## 355          1               1
## 356          1               1
## 357          1               1
## 358          1               0
## 359          1               1
## 360          1               0
## 361          1               1
## 362          1               1
## 363          1               1
## 364          1               0
## 365          1               1
## 366          1               1
## 367          1               1
## 368          1               1
## 369          1               0
## 370          1               1
## 371          1               1
## 374          0               1
## 375          1               0
## 376          1               1
## 378          1               0
## 380          1               1
## 381          1               0
## 385          1               1
## 386       1          1
## 387       1          1
## 389       1          1
## 391       1          1
## 393       1          1
## 395       1          0
## 396       1          1
## 397       0          1
## 398       0          1
## 399       0          0
## 400       0          1
```

Although the algorithm is correct, we can see the age and salary are not good predictors of if the customer purchased from social media ads. This is most likely due to multicollinearity between age and salary. We would need to drop one of the predictors and find a new new one.