Notebook 2 Classification

Matthew Naughton

CS 4375.003

Portfolio: Kernel and Ensemble Methods

```
# Load necessary libraries:
# install.packages("ggplot2") ##uncomment and run if not installed
# install.packages("e1071")   ##uncomment and run if not installed
library(ggplot2)
library(e1071)

set.seed(1234)
df <- read.csv(file = 'housing.csv')

# Divide the data into train and test 75/25
i <- sample(1:nrow(df), nrow(df)*0.75, replace=FALSE)
df$ocean_proximity <- as.factor(df$ocean_proximity)
train <- df[i,]
test <- df[-i,]

# Explore the data
head(df)

##   longitude latitude housing_median_age total_rooms total_bedrooms
population
## 1   -122.23    37.88                 41         880            129
322
## 2   -122.22    37.86                 21        7099           1106
2401
## 3   -122.24    37.85                 52        1467            190
496
## 4   -122.25    37.85                 52        1274            235
558
## 5   -122.25    37.85                 52        1627            280
565
## 6   -122.25    37.85                 52         919            213
413
##   households median_income median_house_value ocean_proximity
## 1        126        8.3252             452600        NEAR BAY
## 2       1138        8.3014             358500        NEAR BAY
## 3        177        7.2574             352100        NEAR BAY
## 4        219        5.6431             341300        NEAR BAY
## 5        259        3.8462             342200        NEAR BAY
## 6        193        4.0368             269700        NEAR BAY

tail(df)
```

```
##       longitude latitude housing_median_age total_rooms total_bedrooms
## 20635   -121.56    39.27                 28        2332            395
## 20636   -121.09    39.48                 25        1665            374
## 20637   -121.21    39.49                 18         697            150
## 20638   -121.22    39.43                 17        2254            485
## 20639   -121.32    39.43                 18        1860            409
## 20640   -121.24    39.37                 16        2785            616
##       population households median_income median_house_value
ocean_proximity
## 20635       1041        344        3.7125             116800
INLAND
## 20636        845        330        1.5603              78100
INLAND
## 20637        356        114        2.5568              77100
INLAND
## 20638       1007        433        1.7000              92300
INLAND
## 20639        741        349        1.8672              84700
INLAND
## 20640       1387        530        2.3886              89400
INLAND
```

```
names(df)
```

```
##  [1] "longitude"          "latitude"           "housing_median_age"
##  [4] "total_rooms"        "total_bedrooms"     "population"
##  [7] "households"         "median_income"      "median_house_value"
## [10] "ocean_proximity"
```

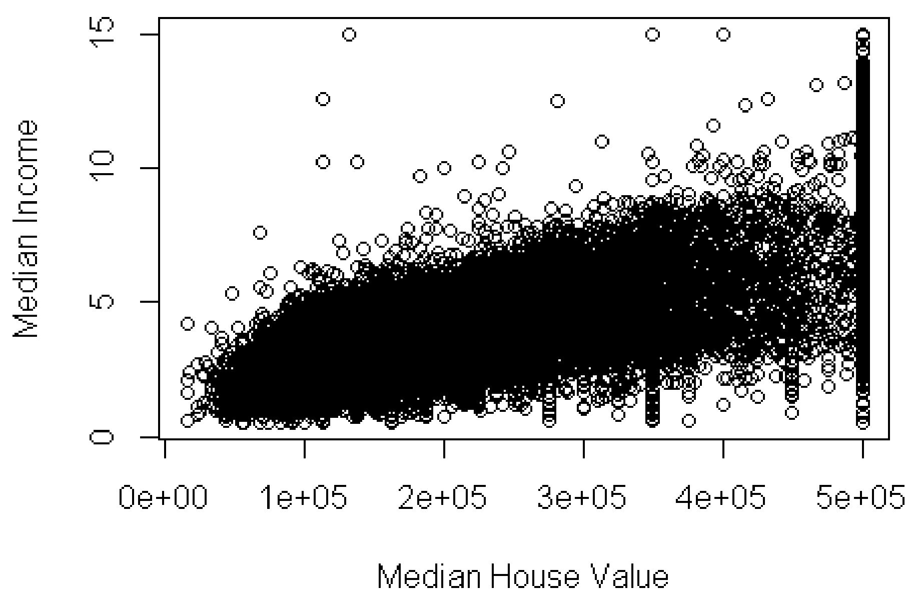
```
str(df)
```

```
## 'data.frame':    20640 obs. of  10 variables:
##  $ longitude         : num  -122 -122 -122 -122 -122 ...
##  $ latitude          : num  37.9 37.9 37.9 37.9 37.9 ...
##  $ housing_median_age: num  41 21 52 52 52 52 52 52 42 52 ...
##  $ total_rooms       : num  880 7099 1467 1274 1627 ...
##  $ total_bedrooms    : num  129 1106 190 235 280 ...
##  $ population        : num  322 2401 496 558 565 ...
##  $ households        : num  126 1138 177 219 259 ...
##  $ median_income     : num  8.33 8.3 7.26 5.64 3.85 ...
##  $ median_house_value: num  452600 358500 352100 341300 342200 ...
##  $ ocean_proximity   : Factor w/ 5 levels "<1H OCEAN","INLAND",..: 4 4 4 4
4 4 4 4 4 4 ...
```

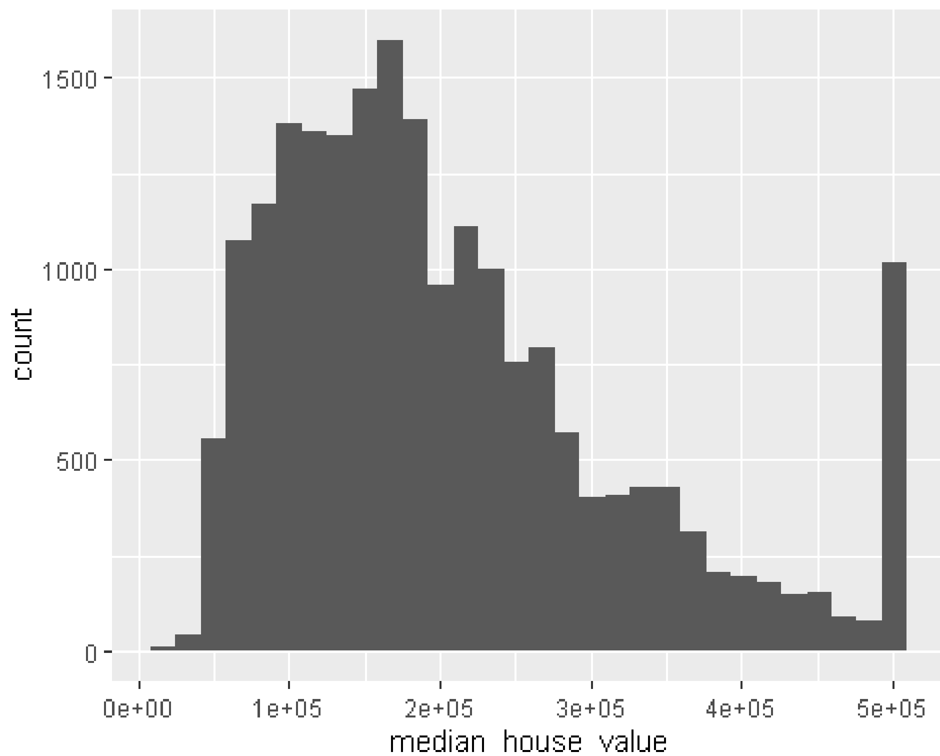
```
summary(df)
```

```
##    longitude         latitude     housing_median_age  total_rooms
##  Min.   :-124.3   Min.   :32.54   Min.   : 1.00      Min.   :    2
##  1st Qu.:-121.8   1st Qu.:33.93   1st Qu.:18.00      1st Qu.: 1448
##  Median :-118.5   Median :34.26   Median :29.00      Median : 2127
##  Mean   :-119.6   Mean   :35.63   Mean   :28.64      Mean   : 2636
```

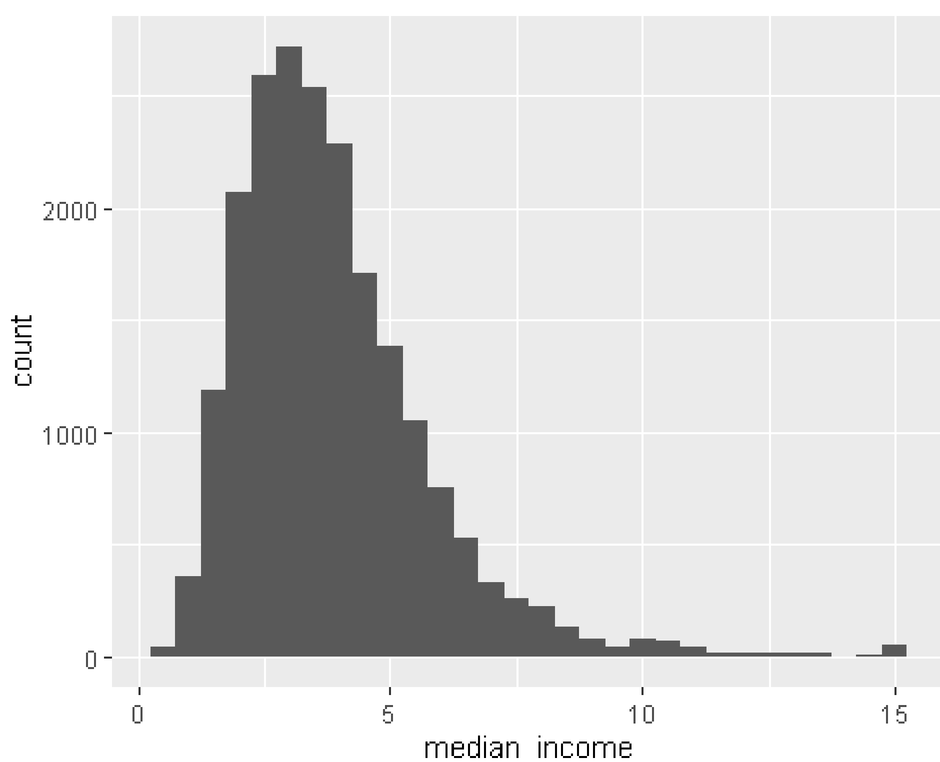
```
##   3rd Qu.:-118.0   3rd Qu.:37.71   3rd Qu.:37.00      3rd Qu.: 3148
##   Max.   :-114.3   Max.   :41.95   Max.   :52.00      Max.   :39320
##
##   total_bedrooms     population      households     median_income
##   Min.   :   1.0   Min.   :     3   Min.   :   1.0   Min.   : 0.4999
##   1st Qu.: 296.0   1st Qu.:   787   1st Qu.: 280.0   1st Qu.: 2.5634
##   Median : 435.0   Median : 1166   Median : 409.0   Median : 3.5348
##   Mean   : 537.9   Mean   : 1425   Mean   : 499.5   Mean   : 3.8707
##   3rd Qu.: 647.0   3rd Qu.: 1725   3rd Qu.: 605.0   3rd Qu.: 4.7432
##   Max.   :6445.0   Max.   :35682   Max.   :6082.0   Max.   :15.0001
##   NA's   :207
##   median_house_value   ocean_proximity
##   Min.   : 14999     <1H OCEAN :9136
##   1st Qu.:119600     INLAND    :6551
##   Median :179700     ISLAND    :   5
##   Mean   :206856     NEAR BAY  :2290
##   3rd Qu.:264725     NEAR OCEAN:2658
##   Max.   :500001
##

plot(df$median_house_value, df$median_income, xlab="Median House Value",
ylab="Median Income")
```



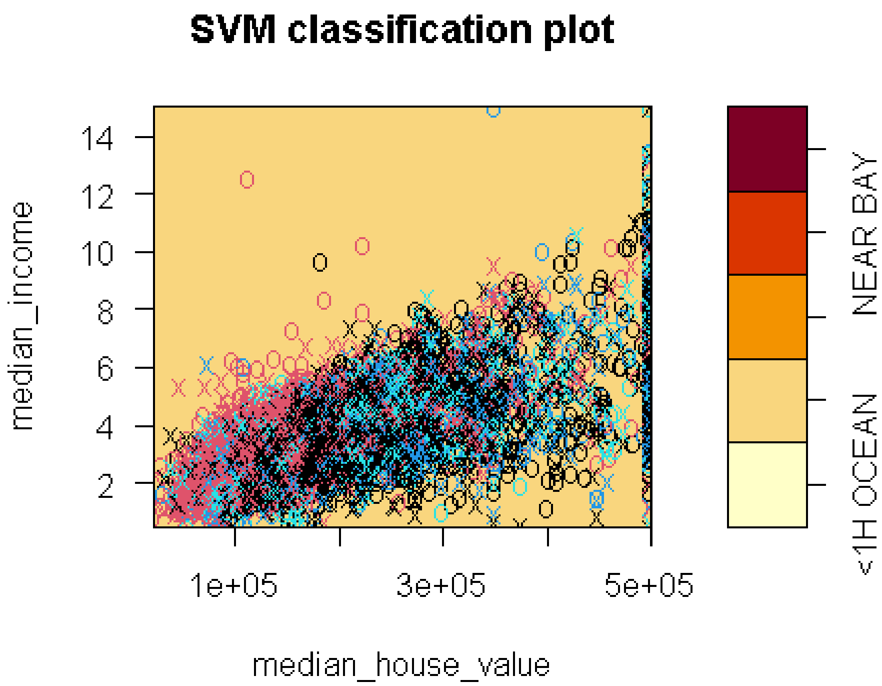

```
ggplot(data=df)+geom_histogram(mapping = aes(x=median_house_value))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(data=df)+geom_histogram(mapping = aes(x=median_income))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

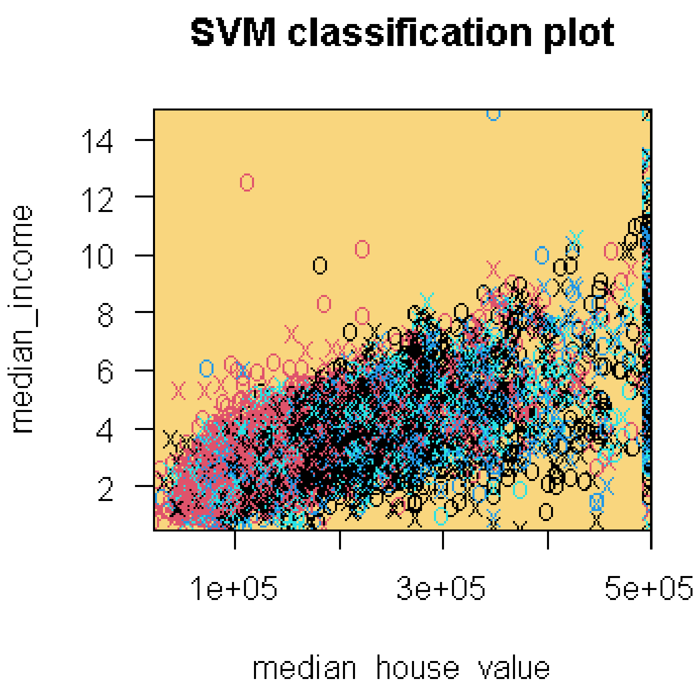


```r
# SVM linear
svm1 <- svm(ocean_proximity~., data=train, kernel="linear", cost=10,
scale=TRUE)
summary(svm1)

##
## Call:
## svm(formula = ocean_proximity ~ ., data = train, kernel = "linear",
##     cost = 10, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##
## Number of Support Vectors:  6644
##
##  ( 2962 1932 659 1086 5 )
##
##
## Number of Classes:  5
##
## Levels:
##  <1H OCEAN INLAND ISLAND NEAR BAY NEAR OCEAN

# Evaluate and plot linear svm
pred <- predict(svm1, newdata=test)
table(pred, test$ocean_proximity[(1:length(pred))])

##
## pred         <1H OCEAN INLAND ISLAND NEAR BAY NEAR OCEAN
##   <1H OCEAN       1878    245      0      102        435
##   INLAND           193   1201      0       46         77
##   ISLAND             0      0      0        0          0
##   NEAR BAY         146     87      0      457        105
##   NEAR OCEAN        38     40      0       10         47

mean(pred==test$ocean_proximity[(1:length(pred))])

## [1] 0.7015861

plot(svm1, test, median_income ~ median_house_value)
```




```
# SVM polynomial kernel

svm2 <- svm(ocean_proximity~., data=train, kernel="polynomial", cost=10,
scale=TRUE)
summary(svm2)

##
## Call:
## svm(formula = ocean_proximity ~ ., data = train, kernel = "polynomial",
##     cost = 10, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  6698
##
##  ( 2906 1751 1094 942 5 )
##
##
## Number of Classes:  5
##
```

```
## Levels:
##  <1H OCEAN INLAND ISLAND NEAR BAY NEAR OCEAN

# Evaluate the polynomial kernel

pred2 <- predict(svm2, newdata=test)
table(pred2, test$ocean_proximity[(1:length(pred2))])

##
## pred2        <1H OCEAN INLAND ISLAND NEAR BAY NEAR OCEAN
##   <1H OCEAN       1865    271      0      136        365
##   INLAND           184   1177      0       49         75
##   ISLAND             0      0      0        0          0
##   NEAR BAY         121     74      0      413         43
##   NEAR OCEAN        85     51      0       17        181

mean(pred2==test$ocean_proximity[(1:length(pred2))])

## [1] 0.711964

plot(svm2, test, median_income ~ median_house_value)
```




```
# SVM radial kernel

svm3 <- svm(ocean_proximity~., data=train, kernel="radial", cost=10, gamma=1,
scale=TRUE)
summary(svm3)
```

```
##
## Call:
## svm(formula = ocean_proximity ~ ., data = train, kernel = "radial",
##     cost = 10, gamma = 1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##
## Number of Support Vectors:  6729
##
##  ( 2804 1452 1477 991 5 )
##
##
## Number of Classes:  5
##
## Levels:
##  <1H OCEAN INLAND ISLAND NEAR BAY NEAR OCEAN
```

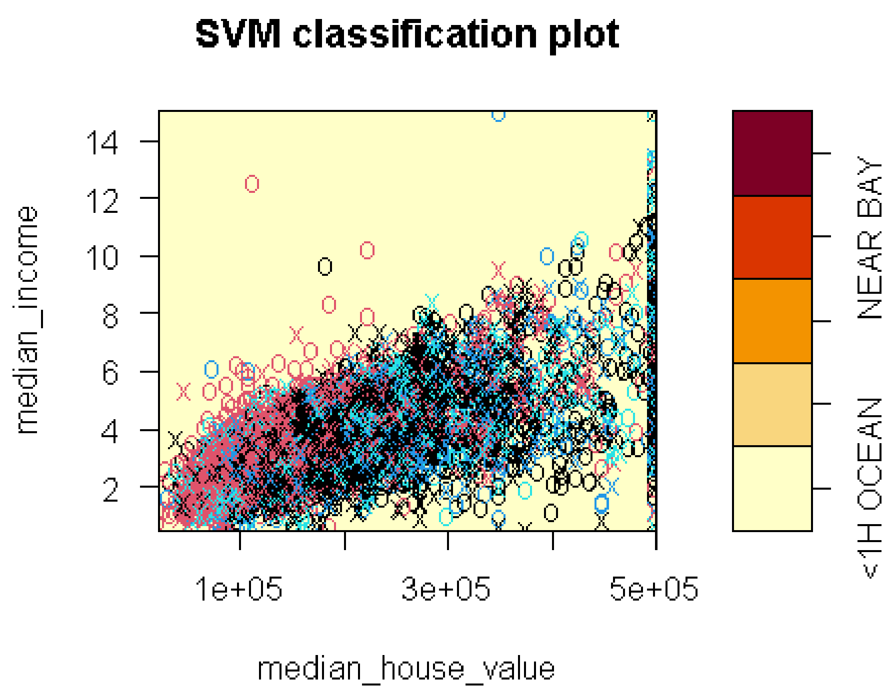
```
# Evaluate radial kernel
pred3 <- predict(svm3, newdata=test)
table(pred3, test$ocean_proximity[(1:length(pred3))])
```

```
##
## pred3        <1H OCEAN INLAND ISLAND NEAR BAY NEAR OCEAN
##   <1H OCEAN       1800    251      0       91        257
##   INLAND           204   1188      0       60         74
##   ISLAND             0      0      0        0          0
##   NEAR BAY          77     51      0      415         43
##   NEAR OCEAN       174     83      0       49        290
```

```
mean(pred3==test$ocean_proximity[(1:length(pred3))])
```

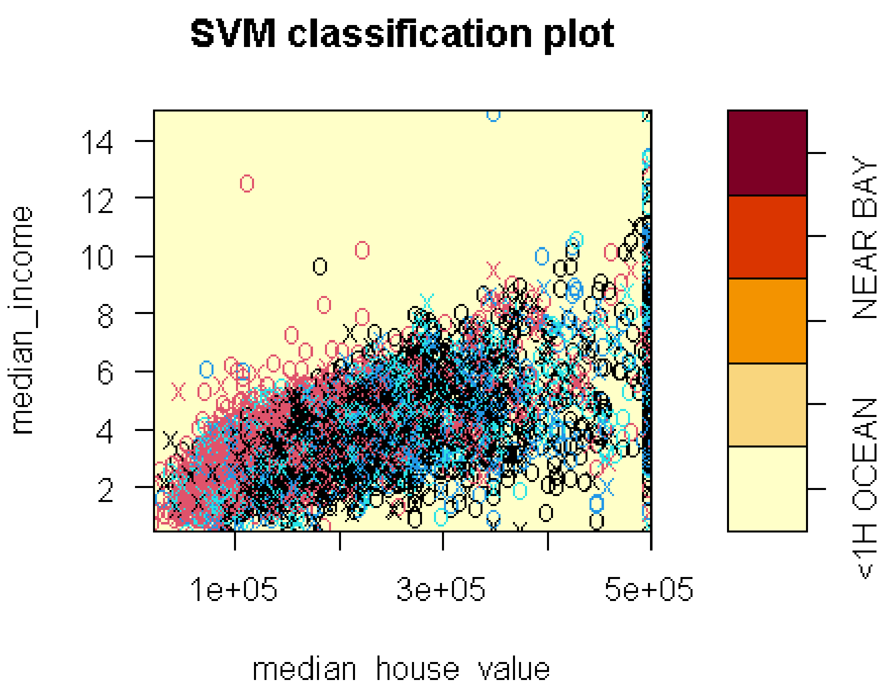
```
## [1] 0.7231251
```

```
plot(svm3, test, median_income ~ median_house_value)
```




```
# Radial kernel with various cost and gamma values
svm4 <- svm(ocean_proximity~., data=train, kernel = "radial", cost=100,
gamma=0.5, scale=TRUE)
summary(svm4)

##
## Call:
## svm(formula = ocean_proximity ~ ., data = train, kernel = "radial",
##     cost = 100, gamma = 0.5, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  100
##
## Number of Support Vectors:  4558
##
##  ( 1866 1104 864 719 5 )
##
##
## Number of Classes:  5
##
## Levels:
##  <1H OCEAN INLAND ISLAND NEAR BAY NEAR OCEAN
```

```r
# Evaluate Radial kernel with various cost/gamma values
pred4 <- predict(svm4, newdata=test)
table(pred4, test$ocean_proximity[(1:length(pred4))])

##
## pred4        <1H OCEAN INLAND ISLAND NEAR BAY NEAR OCEAN
##   <1H OCEAN       1772    245      0       88        257
##   INLAND           221   1200      0       54         73
##   ISLAND             0      0      0        0          0
##   NEAR BAY          71     44      0      428         39
##   NEAR OCEAN       191     84      0       45        295

mean(pred4==test$ocean_proximity[(1:length(pred4))])

## [1] 0.7235167

plot(svm4, test, median_income ~ median_house_value)
```




```r
# d. Provide analysis on why the results were most likely achieved
#The c and gamma values are the biggest modifiers.
#A small c value will create lower bias and high variance.
#A larger gamma value will overfit with low bias and high variance, while a
smaller gamma value could still have higher bias.
#A polynomial kernel will allow us to map multiple lines to get a better fit
#A radial kernel will work better when the data is more clustered.
```