American International University- Bangladesh

Data Science

Final Project Report

Summer 22-23

Name : Nayeem Abdul Qaiyum
ID : 20-43581-1
Section : C

Date Of Submit: 15.08.23

Submitted By

Abdus Salam
Assistant Professor, CS

Data Set Name: US Births 👶 by Year, State, and Education Level

Data set Link: https://www.kaggle.com/datasets/danbraswell/temporary-us-births

Description:

This dataset provides birth rates and related data across the 50 states and DC from 2016 to 2021. A particular emphasis is given to detailed information on the mother's educational level. There are several rows and 9 columns in the data set and they are – State, State.Abbreviation, Year, Gender, Education.Level.of.Mother, Education.Level.Code, Number.of.Births, Average Age.of.Mother..years., Average.Birth.Weight..g. There are different types of attributes in this dataset and they are integer, numeric, character. Here we apply KNN method to find the highl y accurate results.

Table of Contents

Project Solution

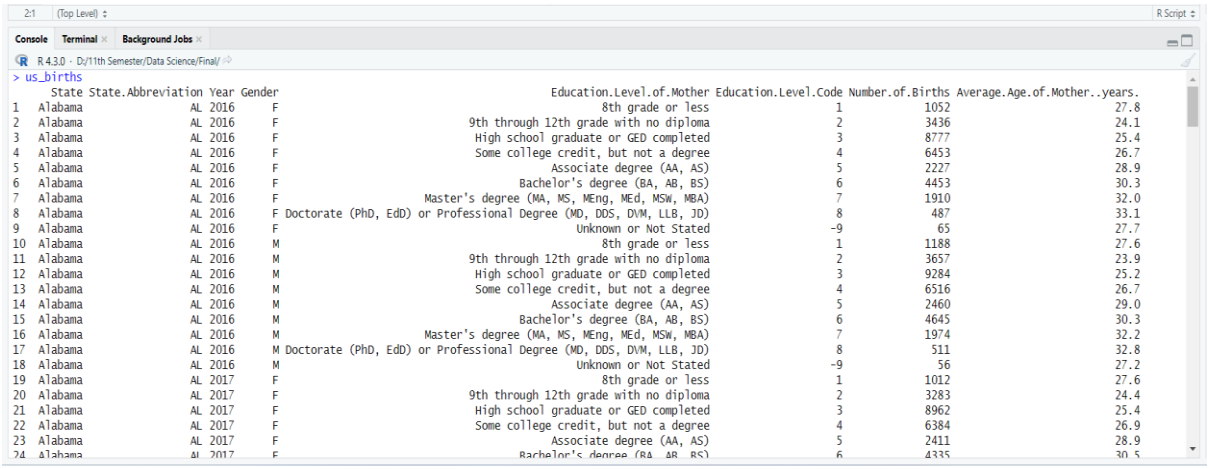
Import data:

Insert all of the data from the excel file first, and then save the document as a dataset file. then convert the dataset file's format to a CSV file. After importing my CSV file into RStudio, I add the following code.

Code Segment:

```
us_births <- read.csv("D:/11th Semester/Data Science/Final/us_births_2016_2021.csv")
us_births
```

Output:



```
> us_births
      State State.Abbreviation Year Gender                                    Education.Level.of.Mother Education.Level.Code Number.of.Births Average.Age.of.Mother..years.
1   Alabama                AL 2016      F                                                8th grade or less                    1             1052                          27.8
2   Alabama                AL 2016      F                                9th through 12th grade with no diploma                    2             3436                          24.1
3   Alabama                AL 2016      F                                 High school graduate or GED completed                    3             8777                          25.4
4   Alabama                AL 2016      F                                 Some college credit, but not a degree                    4             6453                          26.7
5   Alabama                AL 2016      F                                             Associate degree (AA, AS)                    5             2227                          28.9
6   Alabama                AL 2016      F                                       Bachelor's degree (BA, AB, BS)                    6             4453                          30.3
7   Alabama                AL 2016      F                          Master's degree (MA, MS, MEng, MEd, MSW, MBA)                    7             1910                          32.0
8   Alabama                AL 2016      F Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, LLB, JD)                    8              487                          33.1
9   Alabama                AL 2016      F                                             Unknown or Not Stated                   -9               65                          27.7
10  Alabama                AL 2016      M                                                8th grade or less                    1             1188                          27.6
11  Alabama                AL 2016      M                                9th through 12th grade with no diploma                    2             3657                          23.9
12  Alabama                AL 2016      M                                 High school graduate or GED completed                    3             9284                          25.2
13  Alabama                AL 2016      M                                 Some college credit, but not a degree                    4             6516                          26.7
14  Alabama                AL 2016      M                                             Associate degree (AA, AS)                    5             2460                          29.0
15  Alabama                AL 2016      M                                       Bachelor's degree (BA, AB, BS)                    6             4645                          30.3
16  Alabama                AL 2016      M                          Master's degree (MA, MS, MEng, MEd, MSW, MBA)                    7             1974                          32.2
17  Alabama                AL 2016      M Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, LLB, JD)                    8              511                          32.8
18  Alabama                AL 2016      M                                             Unknown or Not Stated                   -9               56                          27.2
19  Alabama                AL 2017      F                                                8th grade or less                    1             1012                          27.6
20  Alabama                AL 2017      F                                9th through 12th grade with no diploma                    2             3283                          24.4
21  Alabama                AL 2017      F                                 High school graduate or GED completed                    3             8962                          25.4
22  Alabama                AL 2017      F                                 Some college credit, but not a degree                    4             6384                          26.9
23  Alabama                AL 2017      F                                             Associate degree (AA, AS)                    5             2411                          28.9
24  Alabama                AL 2017      F                                       Bachelor's degree (BA, AB, BS)                    6             4335                          30.5
```

View the structure of the dataset:

The dataset structure is shown using the str() function, including the variables, their data types, and the initial values. We will get a general idea of the dataset from this.

Code Segment:

```
str(us_births)
```

```
> str(us_births)
'data.frame':   5496 obs. of  9 variables:
 $ State                      : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
 $ State.Abbreviation         : chr  "AL" "AL" "AL" "AL" ...
 $ Year                       : int  2016 2016 2016 2016 2016 2016 2016 2016 2016 2016 ...
 $ Gender                     : chr  "F" "F" "F" "F" ...
 $ Education.Level.of.Mother  : chr  "8th grade or less" "9th through 12th grade with no diploma" "High school graduate or GED completed" "Some college credit, but not a degree" ...
 $ Education.Level.Code       : int  1 2 3 4 5 6 7 8 -9 1 ...
 $ Number.of.Births           : int  1052 3436 8777 6453 2227 4453 1910 487 65 1188 ...
 $ Average.Age.of.Mother..years.: num  27.8 24.1 25.4 26.7 28.9 30.3 32 33.1 27.7 27.6 ...
 $ Average.Birth.Weight..g.   : num  3117 3040 3080 3122 3174 ...
>
```

First few rows of the dataset:

The first few rows of the dataset are shown using the head() function. This will allow us to understand the data and ensure that it was imported properly.

Code Segment:

head(us_births)

```
> head(us_births)
    State State.Abbreviation Year Gender             Education.Level.of.Mother Education.Level.Code Number.of.Births Average.Age.of.Mother..years. Average.Birth.Weight..g.
1 Alabama                AL 2016      F                     8th grade or less                    1            1052                         27.8                    3116.9
2 Alabama                AL 2016      F 9th through 12th grade with no diploma                    2            3436                         24.1                    3040.0
3 Alabama                AL 2016      F  High school graduate or GED completed                    3            8777                         25.4                    3080.0
4 Alabama                AL 2016      F  Some college credit, but not a degree                    4            6453                         26.7                    3121.9
5 Alabama                AL 2016      F             Associate degree (AA, AS)                    5            2227                         28.9                    3174.3
6 Alabama                AL 2016      F         Bachelor's degree (BA, AB, BS)                    6            4453                         30.3                    3239.0
>
```

Column name of the data set:

Explanation: To see the all column name we using the names() function.

Code Segment:

names(us_births)

```
> names(us_births)
[1] "State"                      "State.Abbreviation"         "Year"                       "Gender"                     "Education.Level.of.Mother"
[6] "Education.Level.Code"       "Number.of.Births"           "Average.Age.of.Mother..years." "Average.Birth.Weight..g."
>
```

Find the type of this dataset column:

Explanation: We can determine which column contains which type using sapply().

Code Segment:

sapply(us_births, class)

```
> sapply(us_births, class)
                   State        State.Abbreviation                     Year                   Gender  Education.Level.of.Mother      Education.Level.Code
             "character"              "character"                "integer"                "character"              "character"                 "integer"
       Number.of.Births Average.Age.of.Mother..years.  Average.Birth.Weight..g.
              "integer"                  "numeric"                "numeric"
>
```

Summary:

For numerical variables in the dataset, the summary() function returns summary statistics (count, mean, median, etc.). This will help us gain understanding of the variables' distribution and central patterns.

Code Segment:

summary(us_births)

```
> summary(us_births)
    State          State.Abbreviation      Year        Gender         Education.Level.of.Mother Education.Level.Code Number.of.Births Average.Age.of.Mother..years.
 Length:5496        Length:5496        Min.   :2016   Length:5496        Length:5496              Min.   :-9.000      Min.   :   10    Min.   :23.10
 Class :character   Class :character   1st Qu.:2017   Class :character   Class :character         1st Qu.: 2.000      1st Qu.:  559    1st Qu.:27.50
 Mode  :character   Mode  :character   Median :2019   Mode  :character   Mode  :character         Median : 4.000      Median : 1692    Median :29.60
                                       Mean   :2019                                              Mean   : 3.026      Mean   : 4115    Mean   :29.55
                                       3rd Qu.:2020                                              3rd Qu.: 6.000      3rd Qu.: 5140    3rd Qu.:31.80
                                       Max.   :2021                                              Max.   : 8.000      Max.   :59967    Max.   :35.50

 Average.Birth.Weight..g.
 Min.   :2452
 1st Qu.:3182
 Median :3256
 Mean   :3251
 3rd Qu.:3331
 Max.   :3586
>
```

Data preparation steps

First, I need to prepare my dataset so that I can apply the KNN method later.

To prepare my dataset firstly I need to convert all categorical data to numerical data. Also, we can delete any column unless we need it.

In this dataset I delete one column and that is State Abbreviation.

Delete Column (State Abbreviation):

Code Segment:

```
us_births <- us_births[, -which(names(us_births) == "State.Abbreviation")]

print(us_births)
```

```
> us_births <- us_births[, -which(names(us_births) == "State.Abbreviation")]
> print(us_births)
      State Year Gender                                    Education.Level.of.Mother Education.Level.Code Number.of.Births Average.Age.of.Mother..years.
1   Alabama 2016      F                                            8th grade or less                   1            1052                        27.8
2   Alabama 2016      F                          9th through 12th grade with no diploma                2            3436                        24.1
3   Alabama 2016      F                           High school graduate or GED completed                3            8777                        25.4
4   Alabama 2016      F                           Some college credit, but not a degree                4            6453                        26.7
5   Alabama 2016      F                                    Associate degree (AA, AS)                   5            2227                        28.9
6   Alabama 2016      F                                 Bachelor's degree (BA, AB, BS)                 6            4453                        30.3
7   Alabama 2016      F                     Master's degree (MA, MS, MEng, MEd, MSW, MBA)              7            1910                        32.0
8   Alabama 2016      F Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, LLB, JD)            8             487                        33.1
9   Alabama 2016      F                                          Unknown or Not Stated              -9              65                        27.7
10  Alabama 2016      M                                            8th grade or less                   1            1188                        27.6
11  Alabama 2016      M                          9th through 12th grade with no diploma                2            3657                        23.9
12  Alabama 2016      M                           High school graduate or GED completed                3            9284                        25.2
13  Alabama 2016      M                           Some college credit, but not a degree                4            6516                        26.7
```

Conversion

Converting categorical data to numerical data is a common preprocessing step in data science and analysis. This is often necessary because many algorithms, including K-Nearest Neighbors (KNN), work with numerical data and mathematical calculations.

Categorical to Numeric (State column):

Code Segment:

```
us_births$State<factor(us_births$State,levels=c("Alabama","Alaska","Arizona","Arkansas","California","Colorado","Connecticut","Delaware","District of Columbia","Florida","Georgia","Hawaii","Idaho","Illinois","Indiana","Iowa","Kansas","Kentucky","Louisiana","Maine","Maryland","Massachusetts","Michigan","Minnesota","Mississippi","Missouri","Montana","Nebraska","Nevada","New Hampshire","New
```

Jersey","New Mexico","New York","North Carolina","North Dakota","Ohio","Oklahoma","Oregon","Pennsylvania","Rhode Island","South Carolina","South Dakota","Tennessee","Texas","Utah","Vermont","Virginia","Washington","West Virginia","Wisconsin","Wyoming"), labels = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51))

us_births

```
> us_births$State <- factor(us_births$State, levels=c("Alabama","Alaska","Arizona","Arkansas","California","Colorado","Connecticut","Delaware","District of Columbia","Florida","Georgia","Hawaii","Idaho","Illinois","Indiana","Iowa","Kansas","Kentucky","Louisiana","Maine","Maryland","Massachusetts","Michigan","Minnesota","Mississippi","Missouri","Montana","Nebraska","Nevada","New Hampshire","New Jersey","New Mexico","New York","North Carolina","North Dakota","Ohio","Oklahoma","Oregon","Pennsylvania","Rhode Island","South Carolina","South Dakota","Tennessee","Texas","Utah","Vermont","Virginia","Washington","West Virginia","Wisconsin","Wyoming"), labels = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51))
> us_births
   State Year Gender                                  Education.Level.of.Mother Education.Level.Code Number.of.Births Average.Age.of.Mother..years.
1      1 2016      F                                              8th grade or less                    1            1052                          27.8
2      1 2016      F                                9th through 12th grade with no diploma                    2            3436                          24.1
3      1 2016      F                                 High school graduate or GED completed                    3            8777                          25.4
4      1 2016      F                                 Some college credit, but not a degree                    4            6453                          26.7
5      1 2016      F                                         Associate degree (AA, AS)                    5            2227                          28.9
6      1 2016      F                                      Bachelor's degree (BA, AB, BS)                    6            4453                          30.3
7      1 2016      F                          Master's degree (MA, MS, MEng, MEd, MSW, MBA)                    7            1910                          32.0
8      1 2016      F Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, LLB, JD)                    8             487                          33.1
9      1 2016      F                                          Unknown or Not Stated                   -9              65                          27.7
10     1 2016      M                                              8th grade or less                    1            1188                          27.6
11     1 2016      M                                9th through 12th grade with no diploma                    2            3657                          23.9
12     1 2016      M                                 High school graduate or GED completed                    3            9284                          25.2
13     1 2016      M                                 Some college credit, but not a degree                    4            6516                          26.7
14     1 2016      M                                         Associate degree (AA, AS)                    5            2460                          29.0
15     1 2016      M                                      Bachelor's degree (BA, AB, BS)                    6            4645                          30.3
16     1 2016      M                          Master's degree (MA, MS, MEng, MEd, MSW, MBA)                    7            1974                          32.2
17     1 2016      M Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, LLB, JD)                    8             511                          32.8
18     1 2016      M                                          Unknown or Not Stated                   -9              56                          27.2
```

Categorical to Numeric (Gender column):

Code Segment:

us_births$Gender <- factor(us_births$Gender, levels=c("F","M"), labels = c(1,2))
us_births

```
> us_births$Gender <- factor(us_births$Gender, levels=c("F","M"), labels = c(1,2))
> us_births
   State Year Gender                                  Education.Level.of.Mother Education.Level.Code Number.of.Births Average.Age.of.Mother..years.
1      1 2016      1                                              8th grade or less                    1            1052                          27.8
2      1 2016      1                                9th through 12th grade with no diploma                    2            3436                          24.1
3      1 2016      1                                 High school graduate or GED completed                    3            8777                          25.4
4      1 2016      1                                 Some college credit, but not a degree                    4            6453                          26.7
5      1 2016      1                                         Associate degree (AA, AS)                    5            2227                          28.9
6      1 2016      1                                      Bachelor's degree (BA, AB, BS)                    6            4453                          30.3
7      1 2016      1                          Master's degree (MA, MS, MEng, MEd, MSW, MBA)                    7            1910                          32.0
8      1 2016      1 Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, LLB, JD)                    8             487                          33.1
9      1 2016      1                                          Unknown or Not Stated                   -9              65                          27.7
10     1 2016      2                                              8th grade or less                    1            1188                          27.6
11     1 2016      2                                9th through 12th grade with no diploma                    2            3657                          23.9
12     1 2016      2                                 High school graduate or GED completed                    3            9284                          25.2
13     1 2016      2                                 Some college credit, but not a degree                    4            6516                          26.7
14     1 2016      2                                         Associate degree (AA, AS)                    5            2460                          29.0
15     1 2016      2                                      Bachelor's degree (BA, AB, BS)                    6            4645                          30.3
16     1 2016      2                          Master's degree (MA, MS, MEng, MEd, MSW, MBA)                    7            1974                          32.2
17     1 2016      2 Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, LLB, JD)                    8             511                          32.8
```

Categorical to Numeric (Education Level of Mother column):

Code Segment:

us_births$Education.Level.of.Mother <- factor(us_births$Education.Level.of.Mother, levels=c("8th grade or less","9th through 12th grade with no diploma","High school graduate or GED completed","Some college credit, but not a degree","Associate degree (AA, AS)","Bachelor's degree (BA, AB, BS)","Master's degree (MA, MS, MEng, MEd, MSW,

MBA)","Doctorate (PhD, EdD) or Professional Degree (MD, DDS, DVM, LLB, JD)","Unknown or Not Stated"), labels = c(1,2,3,4,5,6,7,8,9))

us_births

```
> us_births$Education.Level.of.Mother <- factor(us_births$Education.Level.of.Mother, levels=c("8th grade or less","9th through 12th grade with no diploma","High school graduate or GED c
ompleted","Some college credit, but not a degree","Associate degree (AA, AS)","Bachelor's degree (BA, AB, BS)","Master's degree (MA, MS, MEng, MEd, MSW, MBA)","Doctorate (PhD, EdD) or P
rofessional Degree (MD, DDS, DVM, LLB, JD)","Unknown or Not Stated"), labels = c(1,2,3,4,5,6,7,8,9))
> us_births
    State Year Gender Education.Level.of.Mother Education.Level.Code Number.of.Births Average.Age.of.Mother..years. Average.Birth.Weight..g.
1       1 2016      1                        1                    1              1052                          27.8                    3116.9
2       1 2016      1                        2                    2              3436                          24.1                    3040.0
3       1 2016      1                        3                    3              8777                          25.4                    3080.0
4       1 2016      1                        4                    4              6453                          26.7                    3121.9
5       1 2016      1                        5                    5              2227                          28.9                    3174.3
6       1 2016      1                        6                    6              4453                          30.3                    3239.0
7       1 2016      1                        7                    7              1910                          32.0                    3263.5
8       1 2016      1                        8                    8               487                          33.1                    3196.7
9       1 2016      1                        9                   -9                65                          27.7                    3083.9
10      1 2016      2                        1                    1              1188                          27.6                    3232.9
11      1 2016      2                        2                    2              3657                          23.9                    3121.2
12      1 2016      2                        3                    3              9284                          25.2                    3197.9
13      1 2016      2                        4                    4              6516                          26.7                    3252.1
14      1 2016      2                        5                    5              2460                          29.0                    3301.4
15      1 2016      2                        6                    6              4645                          30.3                    3376.1
16      1 2016      2                        7                    7              1974                          32.2                    3358.2
17      1 2016      2                        8                    8               511                          32.8                    3368.4
18      1 2016      2                        9                   -9                56                          27.2                    3107.7
19      1 2017      1                        1                    1              1012                          27.6                    3139.6
20      1 2017      1                        2                    2              3283                          24.4                    3040.6
21      1 2017      1                        3                    3              8962                          25.4                    3068.8
22      1 2017      1                        4                    4              6384                          26.9                    3112.3
23      1 2017      1                        5                    5              2411                          28.9                    3197.2
```

Finding the missing value for all attributes:

Missing data is crucial for accurate analysis and results.

Code Segment:

number_of_missing_value=colSums(is.na(us_births))

number_of_missing_value

```
> number_of_missing_value=colSums(is.na(us_births))
> number_of_missing_value
                     State                       Year                     Gender    Education.Level.of.Mother        Education.Level.Code              Number.of.Births
                         0                          0                          0                          0                          0                             0
Average.Age.of.Mother..years.      Average.Birth.Weight..g.
                         0                             0
>
```

Normalization

Normalization is a data preprocessing technique that is commonly used in data science to scale and transform features to a consistent range (0,1). It involves adjusting the values of features in a dataset to ensure that they have similar scales.

Code Segment:

```
library(dplyr)
us_births <- as.data.frame(sapply(us_births, as.numeric))
min_max_norm <- function(x) {
  (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))
}
```

```
normalized_data <- us_births %>%
  mutate(across(everything(), min_max_norm))
print(normalized_data)
```

```
> library(dplyr)
> us_births <- as.data.frame(sapply(us_births, as.numeric))
> min_max_norm <- function(x) {
+   (x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))
+ }
>
> normalized_data <- us_births %>%
+   mutate(across(everything(), min_max_norm))
> print(normalized_data)
    State Year Gender Education.Level.of.Mother Education.Level.Code Number.of.Births Average.Age.of.Mother..years. Average.Birth.Weight..g.
1    0.00  0.0      0                    0.000           0.5882353     0.0173791217                  0.37903226               0.5865232
2    0.00  0.0      0                    0.125           0.6470588     0.0571409510                  0.08064516               0.5186982
3    0.00  0.0      0                    0.250           0.7058824     0.1462214587                  0.18548387               0.5539778
4    0.00  0.0      0                    0.375           0.7647059     0.1074603466                  0.29032258               0.5909331
5    0.00  0.0      0                    0.500           0.8235294     0.0369764998                  0.46774194               0.6371494
6    0.00  0.0      0                    0.625           0.8823529     0.0741031072                  0.58064516               0.6942141
7    0.00  0.0      0                    0.750           0.9411765     0.0316893774                  0.71774194               0.7158229
8    0.00  0.0      0                    0.875           1.0000000     0.0079557016                  0.80645161               0.6569060
9    0.00  0.0      0                    1.000           0.0000000     0.0009173241                  0.37096774               0.5574175
10   0.00  0.0      1                    0.000           0.5882353     0.0196474140                  0.36290323               0.6888340
11   0.00  0.0      1                    0.125           0.6470588     0.0608269260                  0.06451613               0.5903158
12   0.00  0.0      1                    0.250           0.7058824     0.1546775189                  0.16935484               0.6579644
13   0.00  0.0      1                    0.375           0.7647059     0.1085110996                  0.29032258               0.7057682
14   0.00  0.0      1                    0.500           0.8235294     0.0408626182                  0.47580645               0.7492503
15   0.00  0.0      1                    0.625           0.8823529     0.0773054022                  0.58064516               0.8151349
```

Correlation

Correlation analysis is a statistical technique used to evaluate the strength and direction of the linear relationship between two or more variables in a dataset.

Calculate the correlation between "Education.Level.of.Mother" and "State":

Code Segment:

```
correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$State)
print(correlation)
```

```
> correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$State)
> correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$State)
> print(correlation)
[1] 9.600574e-05
>
```

Calculate the correlation between "Education.Level.of.Mother" and "Year":

Code Segment:

```
correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$Year)
print(correlation)
```

```
> correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$Year)
> print(correlation)
[1] 0.0006628243
>
```

Calculate the correlation between "Education.Level.of.Mother" and "Gender":

Code Segment:

```
correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$Gender)
print(correlation)
```

```
> correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$Gender)
> print(correlation)
[1] 0.0005658527
>
```

Calculate the correlation between "Education.Level.of.Mother" and "Number.of.Births":

Code Segment:

```
correlation <- cor(normalized_data$Education.Level.of.Mother,
normalized_data$Number.of.Births)
print(correlation)
```

```
> correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$Number.of.Births)
> print(correlation)
[1] -0.1347495
>
```

Calculate the correlation between "Education.Level.of.Mother" and "Average.Age.of.Mother..years.":

Code Segment:

```
correlation <- cor(normalized_data$Education.Level.of.Mother,
normalized_data$Average.Age.of.Mother..years.)
print(correlation)
```

```
> correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$Average.Age.of.Mother..years.)
> print(correlation)
[1] 0.6441881
>
```

Calculate the correlation between "Education.Level.of.Mother" and "Average.Birth.Weight..g.":

Code Segment:

```
correlation <- cor(normalized_data$Education.Level.of.Mother,
normalized_data$Average.Birth.Weight..g.)
print(correlation)
```

```
> correlation <- cor(normalized_data$Education.Level.of.Mother, normalized_data$Average.Birth.Weight..g.)
> print(correlation)
[1] 0.08728431
>
```

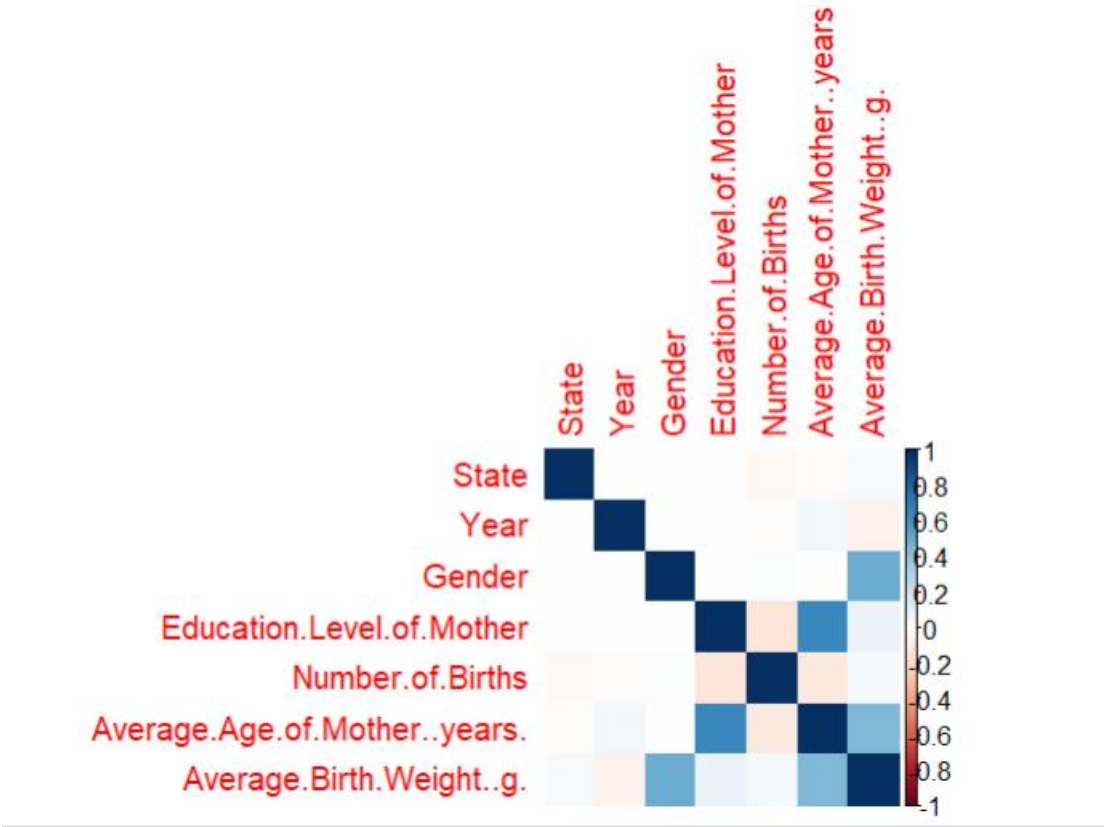
Plot Correlation Matrix

A plot correlation matrix is a data visualization technique that visually represents relationships between multiple variables in a dataset. It displays correlation coefficients between pairs of variables, with color or shading indicating strength and direction. Each cell in the matrix represents the correlation between two variables, and the color or shading of the cell can be used to convey the strength and direction of the correlation.

Code Segment:

```
install.packages("corrplot")

library(corrplot)
plot<-cor(normalized_data)
corrplot(plot,method="color")
```



Training & Testing

Splitting a dataset into training and testing subsets is a crucial step in the field of data science, particularly when building and evaluating predictive models. For example: Fair Comparison, Decision Making, Validation of Results, Quality Control.

Dividing the data into training and test set.

Code Segment:

```
random <- sample(1:nrow(normalized_data), 0.7 * nrow(normalized_data))

# Divide the data set into training and testing sets
```

```
Education.Level.of.Mother_train <- normalized_data[random, ]
Education.Level.of.Mother_test <- normalized_data[-random, ]

# Extract the labels (assuming "Education.Level.of.Mother" column is the label)
Education.Level.of.Mother_train_labels <-
Education.Level.of.Mother_train$Education.Level.of.Mother
Education.Level.of.Mother_test_labels <-
Education.Level.of.Mother_test$Education.Level.of.Mother

Education.Level.of.Mother_train
Education.Level.of.Mother_test
```

For train:

```
> Education.Level.of.Mother_train
     State Year Gender Education.Level.of.Mother Number.of.Births Average.Age.of.Mother..years. Average.Birth.Weight..g.
926   0.16  0.6      0                    0.875     0.0081725236                     0.98387097                0.7481037
1859  0.34  0.2      0                    0.500     0.0393115066                     0.46774194                0.6806315
2445  0.44  0.6      1                    1.000     0.0089897760                     0.47580645                0.5814077
4333  0.80  0.2      0                    0.750     0.0353586737                     0.75806452                0.7166167
1774  0.32  0.4      1                    0.000     0.0088563470                     0.51612903                0.7782678
624   0.10  0.8      1                    0.250     0.1078439548                     0.28225806                0.6531134
4734  0.86  1.0      1                    0.250     0.9248628184                     0.25806452                0.6945669
5150  0.94  0.8      1                    0.500     0.0680654469                     0.57258065                0.8616158
3467  0.64  0.2      0                    0.500     0.1475223910                     0.58870968                0.6766626
284   0.04  0.6      1                    0.500     0.0568574145                     0.54032258                0.8086082
595   0.10  0.6      0                    0.000     0.0121086779                     0.58870968                0.5897866
2355  0.42  0.8      1                    1.000     0.0150607936                     0.69354839                0.7587758
2803  0.50  1.0      1                    0.750     0.0591924212                     0.75000000                0.8501499
992   0.18  0.2      0                    0.125     0.1647847624                     0.16935484                0.5553008
47    0.00  0.4      1                    0.125     0.0521707224                     0.08870968                0.5806139
3664  0.68  0.0      0                    0.375     0.0192638057                     0.35483871                0.7575410
4255  0.78  0.6      0                    0.000     0.0022682923                     0.64516129                0.6219792
2708  0.50  0.0      1                    0.125     0.0621445369                     0.06451613                0.6518786
4969  0.92  0.0      1                    0.375     0.1560284871                     0.40322581                0.7825895
2008  0.36  0.6      1                    0.000     0.0143436129                     0.36290323                0.6658141
1195  0.22  0.0      0                    0.750     0.0110579248                     0.87903226                0.6178338
3919  0.72  0.4      0                    0.750     0.0190469837                     0.72580645                0.7168813
1136  0.20  0.6      0                    0.125     0.0972363527                     0.13709677                0.5157876
886   0.16  0.2      0                    0.375     0.0111579966                     0.40322581                0.5056447
3500  0.64  0.6      0                    0.125     0.1761762597                     0.31451613                0.5916387
```

For test:

```
> Education.Level.of.Mother_test
    State Year Gender Education.Level.of.Mother Number.of.Births Average.Age.of.Mother..years. Average.Birth.Weight..g.
4    0.00  0.0      0                    0.375     0.1074603466                     0.29032258                0.5909331
5    0.00  0.0      0                    0.500     0.0369764998                     0.46774194                0.6371494
8    0.00  0.0      0                    0.875     0.0079557016                     0.80645161                0.6569060
16   0.00  0.0      1                    0.750     0.0327568090                     0.73387097                0.7993473
17   0.00  0.0      1                    0.875     0.0083559885                     0.78225806                0.8083436
19   0.00  0.2      0                    0.000     0.0167119769                     0.36290323                0.6065444
22   0.00  0.2      0                    0.375     0.1063095218                     0.30645161                0.5824660
26   0.00  0.2      0                    0.875     0.0090231332                     0.80645161                0.6869818
28   0.00  0.2      1                    0.000     0.0161449038                     0.36290323                0.6860998
36   0.00  0.2      1                    1.000     0.0012508965                     0.29032258                0.5634151
39   0.00  0.4      0                    0.250     0.1482896076                     0.19354839                0.5404833
40   0.00  0.4      0                    0.375     0.1003052187                     0.31451613                0.5987829
48   0.00  0.4      1                    0.250     0.1566956319                     0.19354839                0.6392662
53   0.00  0.4      1                    0.875     0.0093400270                     0.80645161                0.7669783
54   0.00  0.4      1                    1.000     0.0008672882                     0.36290323                0.6039866
59   0.00  0.6      0                    0.500     0.0398619010                     0.47580645                0.6413830
61   0.00  0.6      0                    0.750     0.0310055540                     0.70967742                0.6910390
64   0.00  0.6      1                    0.000     0.0200477009                     0.39516129                0.7003881
66   0.00  0.6      1                    0.250     0.1607318578                     0.19354839                0.6417358
80   0.00  0.8      0                    0.875     0.0089897760                     0.81451613                0.6846887
82   0.00  0.8      1                    0.000     0.0188635189                     0.34677419                0.6706650
84   0.00  0.8      1                    0.250     0.1608819654                     0.20967742                0.6353854
86   0.00  0.8      1                    0.500     0.0438480911                     0.49193548                0.7344329
96   0.00  1.0      0                    0.625     0.0745534300                     0.59677419                0.6754278
97   0.00  1.0      0                    0.750     0.0360758544                     0.70967742                0.6711060
98   0.00  1.0      0                    0.875     0.0094567774                     0.79838710                0.7169695
103  0.00  1.0      1                    0.375     0.0945344163                     0.34677419                0.6842477
```

Accuracy:

In data science and machine learning, accuracy is a key metric used to measure the performance of a predictive model.

Code Segment:

```
library(class)
set.seed(123)
random <- sample(1:nrow(normalized_data), 0.7 * nrow(normalized_data))
Education.Level.of.Mother_train <- normalized_data[random, ]
Education.Level.of.Mother_test <- normalized_data[-random, ]
Education.Level.of.Mother_train_labels <-
Education.Level.of.Mother_train$Education.Level.of.Mother
Education.Level.of.Mother_test_labels <-
Education.Level.of.Mother_test$Education.Level.of.Mother

  k <- 3
predicted_labels <- knn(train = Education.Level.of.Mother_train[, -
which(names(Education.Level.of.Mother_train) == "Education.Level.of.Mother")],
                        test = Education.Level.of.Mother_test[, -
which(names(Education.Level.of.Mother_test) == "Education.Level.of.Mother")],
                        cl = Education.Level.of.Mother_train_labels,
                        k = k)

accuracy <- sum(predicted_labels == Education.Level.of.Mother_test_labels) /
length(Education.Level.of.Mother_test_labels)

cat("Accuracy:", accuracy, "\n")
```

```
> library(class)
> set.seed(123)
> random <- sample(1:nrow(normalized_data), 0.7 * nrow(normalized_data))
> Education.Level.of.Mother_train <- normalized_data[random, ]
> Education.Level.of.Mother_test <- normalized_data[-random, ]
> Education.Level.of.Mother_train_labels <- Education.Level.of.Mother_train$Education.Level.of.Mother
> Education.Level.of.Mother_test_labels <- Education.Level.of.Mother_test$Education.Level.of.Mother
>
>   k <- 3
> predicted_labels <- knn(train = Education.Level.of.Mother_train[, -which(names(Education.Level.of.Mother_train) == "Education.Level.of.Moth
er")],
+                         test = Education.Level.of.Mother_test[, -which(names(Education.Level.of.Mother_test) == "Education.Level.of.Mothe
r")],
+                         cl = Education.Level.of.Mother_train_labels,
+                         k = k)
>
> accuracy <- sum(predicted_labels == Education.Level.of.Mother_test_labels) / length(Education.Level.of.Mother_test_labels)
>
> cat("Accuracy:", accuracy, "\n")
Accuracy: 0.4893875
>
```

Dividing the data into training and test set

```
library(class)

# Split the data into training and test sets
set.seed(123)
random <- sample(1:nrow(normalized_data), 0.7 * nrow(normalized_data))
train_data <- normalized_data[random, ]
test_data <- normalized_data[-random, ]

# Extract labels
train_labels <- train_data$Education.Level.of.Mother
test_labels <- test_data$Education.Level.of.Mother

# Define k value
k <- 3

# Train KNN classifier
knn_model <- knn(train = train_data[, -which(names(train_data) ==
"Education.Level.of.Mother")],
                 test = test_data[, -which(names(test_data) == "Education.Level.of.Mother")],
                 cl = train_labels,
                 k = k)

# Calculate accuracy
accuracy_approach1 <- sum(knn_model == test_labels) / length(test_labels)
cat("Accuracy (Dividing data into training and test sets):", accuracy_approach1, "\n")
```

```
+                 test = test_data[, -which(names(test_data) == "Education.Level.of.Mother")],
+                 cl = train_labels,
+                 k = k)
>
> # Calculate accuracy
> accuracy_approach1 <- sum(knn_model == test_labels) / length(test_labels)
> cat("Accuracy (Dividing data into training and test sets):", accuracy_approach1, "\n")
Accuracy (Dividing data into training and test sets): 0.4893875
>
```

10-fold cross validation

The 10-fold cross-validation method, which divides the dataset into 10 equal-sized subsets, is a common data science method for evaluating the effectiveness of predictive models. Its main goal is to give an accurate estimate of how well a model performs on unknown data.

Code Segment:

```
install.packages("class")
install.packages("caret")
```

```
library(class)
library(caret)

set.seed(123)

num_folds <- 10

fold_indices <- createFolds(normalized_data$Education.Level.of.Mother, k = num_folds)

accuracies <- numeric(num_folds)

for (i in 1:num_folds) {

  test_indices <- fold_indices[[i]]

  train_indices <- setdiff(1:nrow(normalized_data), test_indices)
  Education.Level.of.Mother_train <- normalized_data[train_indices, ]

  Education.Level.of.Mother_test <- normalized_data[test_indices, ]

  input_features_train <- Education.Level.of.Mother_train[, c("State", "Year",
"Gender","Number.of.Births", "Average.Age.of.Mother..years.",
"Average.Birth.Weight..g.")]

  input_features_test <- Education.Level.of.Mother_test[, c("State", "Year",
"Gender","Number.of.Births", "Average.Age.of.Mother..years.",
"Average.Birth.Weight..g.")]

  Education.Level.of.Mother_train_labels <-
Education.Level.of.Mother_train$Education.Level.of.Mother

  Education.Level.of.Mother_test_labels <-
Education.Level.of.Mother_test$Education.Level.of.Mother


  k <- 3  # Set the value of 'k'

  predicted_labels <- knn(train = input_features_train,

                          test = input_features_test,

                          cl = Education.Level.of.Mother_train_labels,
```

```
                    k = k)

  accuracies[i] <- sum(predicted_labels == Education.Level.of.Mother_test_labels) /
length(Education.Level.of.Mother_test_labels)

}

mean_accuracy <- mean(accuracies)

cat("Mean Accuracy (10-Fold Cross-Validation):", mean_accuracy, "\n")
'
+ }
>
>
>
> mean_accuracy <- mean(accuracies)
>
> cat("Mean Accuracy (10-Fold Cross-Validation):", mean_accuracy, "\n")
Mean Accuracy (10-Fold Cross-Validation): 0.5151042
```

Confusion matrix

A confusion matrix evaluates classification model performance by comparing predicted and actual classes, revealing strengths and weaknesses, and aiding in data science.

Code Segment:

```
library(class)
library(caret)

# Set seed for reproducibility
set.seed(123)

# Assuming 'normalized_data' is your original dataset
# Replace this with the correct name if necessary

# Number of folds for cross-validation
num_folds <- 10

# Create indices for cross-validation folds
fold_indices <- createFolds(normalized_data$Education.Level.of.Mother, k = num_folds)

# Initialize matrices to store confusion matrices and metrics
confusion_matrices <- list()
recalls <- numeric(num_folds)
```

```
precisions <- numeric(num_folds)

# Define a function to calculate recall and precision
calculate_metrics <- function(cm) {
  recall <- cm[1, 1] / sum(cm[1, ])
  precision <- cm[1, 1] / sum(cm[, 1])
  return(list(recall = recall, precision = precision))
}

# Perform 10-fold cross-validation
for (i in 1:num_folds) {
  # Split data into training and testing sets for this fold
  test_indices <- fold_indices[[i]]
  train_indices <- setdiff(1:nrow(normalized_data), test_indices)

  data_train <- normalized_data[train_indices, ]
  data_test <- normalized_data[test_indices, ]

  # Extract the input features and the decision attribute
  input_features_train <- data_train[, c("State", "Year", "Gender", "Number.of.Births",
                                         "Average.Age.of.Mother..years.", "Average.Birth.Weight..g.")]
  input_features_test <- data_test[, c("State", "Year", "Gender", "Number.of.Births",
                                       "Average.Age.of.Mother..years.", "Average.Birth.Weight..g.")]
  decision_train <- data_train$Education.Level.of.Mother
  decision_test <- data_test$Education.Level.of.Mother

  # Perform KNN classification
  k <- 3  # Set the value of 'k'
  predicted_decisions <- knn(train = input_features_train,
                             test = input_features_test,
                             cl = decision_train,
                             k = k)

  # Calculate confusion matrix for this fold
  confusion_matrices[[i]] <- table(predicted = predicted_decisions, actual = decision_test)

  # Calculate recall and precision for this fold
  metrics <- calculate_metrics(confusion_matrices[[i]])
  recalls[i] <- metrics$recall
  precisions[i] <- metrics$precision
}

# Calculate the mean recall and precision across folds
```

```
mean_recall <- mean(recalls)
mean_precision <- mean(precisions)
# Print mean recall and precision
cat("Mean Recall:", mean_recall, "\n")
cat("Mean Precision:", mean_precision, "\n")

# Print individual confusion matrices for each fold
for (i in 1:num_folds) {
  cat("Confusion Matrix (Fold", i, "):\n")
  print(confusion_matrices[[i]])
  cat("\n")
}
```

```
> mean_precision <- mean(precisions)
>
> # Print mean recall and precision
> cat("Mean Recall:", mean_recall, "\n")
Mean Recall: 0.2494325
> cat("Mean Precision:", mean_precision, "\n")
Mean Precision: 0.281739
>
> # Print individual confusion matrices for each fold
> for (i in 1:num_folds) {
+   cat("Confusion Matrix (Fold", i, "):\n")
+   print(confusion_matrices[[i]])
```

Confusion Matrix (1,2)

```
Confusion Matrix (Fold 1 ):
          actual
predicted  0 0.125 0.25 0.375 0.5 0.625 0.75 0.875  1
    0     16     1    0     9  24     5    1     0 19
    0.125  3    44   17     3   0     0    0     0  0
    0.25   2    13   34    10   1     0    0     0  0
    0.375  5     2   10    24   5     0    0     0  0
    0.5   16     0    0     8  18    10    0     0  6
    0.625  8     0    0     0  11    32    6     1  0
    0.75   1     0    0     0   0    12   29    10  3
    0.875  0     0    0     0   0     0   24    48  2
    1      9     0    2     1   8     1    2     1 32

Confusion Matrix (Fold 2 ):
          actual
predicted  0 0.125 0.25 0.375 0.5 0.625 0.75 0.875  1
    0     18     1    1     7  22     5    0     0 13
    0.125  0    42   14     1   0     0    0     0  2
    0.25   1    18   32    10   1     0    0     0  1
    0.375  6     1   23    39   5     3    0     0  3
    0.5   12     0    0     7  18    13    1     0 11
    0.625  4     0    0     0   5    36    4     0  1
    0.75   2     0    0     0   0     9   27     9  0
    0.875  0     0    0     0   0     0   23    45  1
    1      9     0    0     1   7     2    0     0 36
```

Confusion Matrix (3,4)

Confusion Matrix (Fold 3):

	actual								
predicted	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
0	12	1	1	10	25	4	0	0	10
0.125	4	45	18	0	0	0	0	0	0
0.25	1	13	31	7	0	0	0	0	1
0.375	6	1	17	25	4	1	0	0	5
0.5	20	0	0	5	22	13	0	0	10
0.625	3	0	0	0	13	29	5	0	0
0.75	0	0	0	0	0	9	36	22	0
0.875	0	0	0	0	0	0	23	47	0
1	10	0	1	3	9	2	1	0	26

Confusion Matrix (Fold 4):

	actual								
predicted	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
0	18	1	0	10	22	2	1	0	11
0.125	0	44	13	1	0	0	0	0	0
0.25	0	14	37	11	0	0	0	0	1
0.375	11	0	16	25	4	2	0	0	0
0.5	18	0	0	5	25	11	0	0	8
0.625	1	0	0	1	9	30	6	0	0
0.75	0	0	0	0	0	16	29	14	3
0.875	0	0	0	0	0	0	23	39	0
1	10	0	0	2	7	2	0	0	45

Confusion Matrix (5,6)

Confusion Matrix (Fold 5):

	actual								
predicted	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
0	21	0	0	12	19	3	0	0	17
0.125	1	42	12	2	0	0	0	0	1
0.25	3	15	32	12	1	0	0	0	0
0.375	9	6	9	30	4	1	0	0	3
0.5	19	0	0	2	21	8	0	0	11
0.625	3	0	0	2	10	30	6	0	2
0.75	2	0	0	0	0	19	28	17	1
0.875	0	0	0	0	0	0	23	44	0
1	9	1	0	3	4	3	1	0	25

Confusion Matrix (Fold 6):

	actual								
predicted	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
0	20	1	1	7	21	3	0	0	19
0.125	5	45	11	5	0	0	0	0	1
0.25	1	11	32	11	0	0	0	0	0
0.375	8	2	14	33	2	1	0	0	3
0.5	17	0	0	4	16	15	0	0	6
0.625	6	0	0	0	13	31	5	1	3
0.75	0	0	0	0	0	2	36	8	1
0.875	1	0	0	0	0	1	27	47	3
1	8	0	1	2	8	0	1	0	29

Confusion Matrix (7,8)

Confusion Matrix (Fold 7):

	actual								
predicted	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
0	24	0	0	8	27	6	0	0	9
0.125	2	42	12	3	0	0	0	0	2
0.25	1	10	33	9	0	0	0	0	1
0.375	5	1	15	31	2	1	0	0	2
0.5	15	1	1	5	18	13	1	0	7
0.625	5	0	0	0	8	29	16	0	2
0.75	0	0	0	0	0	13	23	12	2
0.875	0	0	0	0	0	0	20	54	0
1	16	0	0	8	3	0	1	0	30

Confusion Matrix (Fold 8):

	actual								
predicted	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
0	12	1	2	16	20	1	1	0	19
0.125	1	54	15	2	0	0	0	0	0
0.25	4	12	26	4	1	0	0	0	1
0.375	6	2	16	30	2	2	0	0	6
0.5	16	0	0	7	25	11	0	0	4
0.625	4	0	0	0	7	23	8	0	0
0.75	2	0	0	0	0	20	36	16	2
0.875	0	0	0	0	0	0	18	43	1
1	9	0	1	3	6	2	1	0	29

Confusion Matrix (9,10)

Confusion Matrix (Fold 9):

	actual								
predicted	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
0	15	1	1	12	19	6	0	0	11
0.125	1	47	19	3	0	0	0	0	0
0.25	0	13	30	10	0	0	0	0	0
0.375	5	2	9	28	2	1	0	0	7
0.5	27	0	0	14	20	10	1	0	6
0.625	5	0	0	0	7	34	8	0	1
0.75	0	0	0	0	0	12	29	12	1
0.875	0	0	0	0	0	1	18	56	1
1	9	0	0	1	6	1	1	0	26

Confusion Matrix (Fold 10):

	actual								
predicted	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
0	17	1	2	11	21	7	2	0	6
0.125	4	48	19	3	0	0	0	0	0
0.25	2	12	20	11	0	0	0	0	0
0.375	4	1	11	33	1	0	0	0	2
0.5	23	0	0	9	16	11	0	0	6
0.625	2	0	0	0	13	27	6	0	1
0.75	2	0	0	0	0	14	27	7	0
0.875	0	0	0	0	0	0	24	59	0
1	15	0	1	1	4	1	3	0	40