```
---
title: "Cancer Mortality Project"
author: "Chenyan Feng"
date: "2023-03-20"
output:
  html_document:
    toc: yes
  word_document:
    toc: yes
  pdf_document:
    toc: yes
---
```

1.Data preprocessing and analysis before fitting model

For this project, our goal is by using CancerData file as training set and CancerHoldoutData file as testing set to understand how different socio-economic factor might influence health and mortality.

From the data file, we can see there are missing data in PctSomeCol18-24 (Percent of county residents ages 18-24 highest education attained: some college) and incorrect data in PctAsian, PctBlack and PctOtherRace(there are a lot "0" in the two columns). I will take place the missing data with mean value take place the "0" with media.

What variable look most promising for predicting cancer mortality from exploratory data analysis and why?

The variables' meanings can be found in the data appendix. In my opinion, the variables that hold the most potential for predicting cancer mortality are median income, poverty percentage, median age, average household size, percent private coverage, percent public coverage, and percent race(medianIncome, povertyPercent, MedianAge, AvgHouseholdSize, PctPrivateCoverage, PctPublicCoverage and the PctRace.). Wealthier individuals may be able to afford medical expenses and decrease their chances of developing cancer, while poverty may hinder individuals' ability to access necessary medical treatments. Age is also an important factor, as a higher age corresponds to a higher probability of developing cancer due to a weakened immune system. Furthermore, larger households may struggle to afford medical expenses, and having medical insurance through private or public coverage may decrease the likelihood of developing cancer by allowing individuals to receive regular check-ups. Finally, genetic factors may also play a role in cancer development, and thus race is an important variable to consider. These variables are what I believe to be the most promising for predicting cancer mortality.

Identify data quality issue in this dataset, Enumerate how it will address the identified data quality issues.

Like I said above, there are missing data in PctSomeCol18-24 (Percent of county residents ages 18-24 highest education attained: some college) and incorrect data in PctAsian, PctBlack and PctOtherRace(there are a lot "0" in the two columns). I will take place the missing data with mean value take place the "0" with media.

Is there any collinearity between variables? Can it be detected? How it address collinearity affects model performance?

Yes, There are many reasons why multicollinearity may occur. For example, there are some high relationship between MedianAge, MedianAgeMale and MedianAgeFemale. Because MedianAge include MedianAgeMale and MedianAgeFemale. So I will use Variance Inflation Factor to see if there are some high multicollinearity between variables. For addressing multicollinearity, we can removing variables or using principal component analysis.

~~~
```{r}
train_data = read.csv("CancerData.csv")
test_data = read.csv("CancerHoldoutData.csv")
# Impute missing values of PctSomeCol18_24 with mean
train_data$PctSomeCol18_24[is.na(train_data$PctSomeCol18_24)] = mean(train_data$PctSomeCol18_24, na.rm = TRUE)
test_data$PctSomeCol18_24[is.na(test_data$PctSomeCol18_24)] = mean(train_data$PctSomeCol18_24, na.rm = TRUE)

# Impute "0" values of PctAsian, PctBlack and PctOtherRace with median
train_data$PctAsian = ifelse(train_data$PctAsian == 0, median(train_data$PctAsian), train_data$PctAsian)
train_data$PctOtherRace = ifelse(train_data$PctOtherRace == 0, median(train_data$PctOtherRace),
train_data$PctOtherRace)
train_data$PctBlack = ifelse(train_data$PctBlack == 0, median(train_data$PctBlack), train_data$PctBlack)
# Apply same code to test set
test_data$PctAsian = ifelse(test_data$PctAsian == 0, median(test_data$PctAsian), test_data$PctAsian)
test_data$PctOtherRace = ifelse(test_data$PctOtherRace == 0, median(test_data$PctOtherRace),
test_data$PctOtherRace)
test_data$PctBlack = ifelse(test_data$PctBlack == 0, median(test_data$PctBlack), test_data$PctBlack)
library(car)
vif(lm(TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent + MedianAge + MedianAgeMale +
MedianAgeFemale + AvgHouseholdSize + PercentMarried + PctNoHS18_24+ PctHS18_24 + PctSomeCol18_24 +
PctBachDeg18_24 + PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone + PctWhite + PctBlack +
PctAsian + PctOtherRace + PctMarriedHouseholds, data = train_data))

```
~~~

From above result, the variables' VIF is larger than 5 are: medIncome, povertyPercent, MedianAgeMale, MedianAgeFemale, PercentMarried, PctPrivateCoverage, PctPublicCoverage, PctPublicCoverageAlone, PctWhite, PctBlack and PctMarriedHouseholds. So those variables are highly multicollinearity. In order to not influence my model fitting, I would prefer to not use those highly multicollinearity variables.

2.Linear Regression

Develop a linear regression model.

To fit a linear regression, I will begin by fitting all variables and subsequently assess their p values. Any variable with a p value less than 0.05 will then be removed.

```{r}
lm.fit = lm(TARGET_deathRate ~ incidenceRate + medIncome + povertyPercent + MedianAge + MedianAgeMale +
MedianAgeFemale + AvgHouseholdSize + PercentMarried + PctNoHS18_24+ PctHS18_24 + PctSomeCol18_24 +
PctBachDeg18_24 + PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone + PctWhite + PctBlack +
PctAsian + PctOtherRace + PctMarriedHouseholds, data = train_data)
```

```{r}
summary(lm.fit)

# Detect multicollinearity

cor(train_data[, c('incidenceRate', 'medIncome', 'PctHS18_24','PctBachDeg18_24', 'PctPrivateCoverage',
'PctPublicCoverageAlone','PctOtherRace')])
```
What variables are significant, insignificant, how does removing variables affect model performance
From the above coding we can know that variables which p value is less than 0.05 are incidenceRate,PctMarriedHouseholds, povertyPercent, medIncome, PctHS18_24, PctBachDeg18_24, PctPrivateCoverage, PctPublicCoverageAlone and PctOtherRace. Therefore those variables are important.

On the other hand, the unimportant variables are:MedianAge, PctNoHS18_24, PctSomeCol18_24, PctPublicCoverage, PctWhite, PctBlack, PctAsian.

After removing the unimportant variables, both the R-squared and adjusted R-squared values exhibited a minor decline from 0.4729 and 0.4688, respectively, to 0.4673 and 0.4656. However, the F-statistic values significantly improved in the last fitting, rising from 115.3 to 283, indicating a significant improvement in our model.

```{r}
# Detect multicollinearity
vif(lm(TARGET_deathRate~incidenceRate+medIncome+PctMarriedHouseholds+
povertyPercent+PctHS18_24+PctBachDeg18_24+PctPrivateCoverage+PctPublicCoverageAlone+PctOtherRace, data =
train_data))
```
Both the VIF and correlation functions indicate that the VIF factors for PctPrivateCoverage and PctPublicCoverageAlones are over 5. As a result, I will attempt to remove PctPrivateCoverage.
```{r}
vif(lm(TARGET_deathRate~incidenceRate+medIncome+PctHS18_24+PctBachDeg18_24+PctMarriedHouseholds+
povertyPercent+PctPublicCoverageAlone+PctOtherRace, data = train_data))
```
After removing the PctprivateCoverage variable, all of the remaining variables' VIF factors are within an acceptable range, which is less than 5.

So, I will use those selected variables (incidenceRate, mdeIncome, PctHS18_24, PctBachEg18_24, PctMarriedHouseholds, povertyPercent, PctpublicCoverageAlone, PctotherRace) to check is there any outliers and fit a new linear regression model.

Present and interpret model diagnosis. What insights obtain to improve the model from diagnosis?

Based on the Residuals vs Fitted plot, there may be some degree of non-linearity, as the trend line is not perfectly straight.

Similarly, the Normal Q-Q plot exhibits an S-shape and seems to deviate above the line, indicating a possible difference in variance rather than equal variance.

Upon reviewing the Residuals vs Leverage plot, it is evident that the points located in row 2314, 1177, and 1053 are pulling away from the linear regression model. These points exhibit high leverage and influence and will be treated as outliers.

```{r}
lm.fit = lm(TARGET_deathRate~incidenceRate+medIncome+PctHS18_24+PctBachDeg18_24+PctMarriedHouseholds+
povertyPercent+PctPublicCoverageAlone+PctOtherRace, data = train_data)
summary(lm.fit)
# diagnostic plots
plot(lm.fit, which = 1) # Residuals vs Fitted
plot(lm.fit, which = 2) # Normal Q-Q
plot(lm.fit, which = 3) # Scale-Location
plot(lm.fit, which = 5) # Residuals vs Leverage
```

Two non-linear and one interaction and evaluation about how they affect model performance and diagnosis.

In order to obtain non-linear and interaction effects, we can square the variables PctHS18_24 and PctPublicCoverageAlone. Additionally, we can create interaction terms between incidenceRate and mdeIncome.

```{r}
```

~~~~
new_df <- train_data[-c(1177,1053,2314), ]
lm.fit = lm(TARGET_deathRate~incidenceRate*medIncome+povertyPercent+PctMarriedHouseholds
+PctHS18_24+I(PctHS18_24^2)+PctBachDeg18_24+PctPublicCoverageAlone+I(PctPublicCoverageAlone^2)+PctOtherRace,
data = new_df)
summary(lm.fit)
# diagnostic plots
plot(lm.fit, which = 1) # Residuals vs Fitted
plot(lm.fit, which = 2) # Normal Q-Q
plot(lm.fit, which = 3) # Scale-Location
plot(lm.fit, which = 5) # Residuals vs Leverage
```

The plots above indicate that the current model is an improvement over the previous fit.

The Residuals vs Fitted plot shows greater linearity, and the Normal Q-Q plot is closer to the straight line.

Additionally, the Residuals vs Leverage plot is better than the previous one, suggesting that there are no
outliers in the current model fit.

Furthermore, it is evident that both the R-squared value (0.4807) and the adjusted R-squared (0.4785) value
have improved in comparison to the previous model.



# 3.KNN model
## Split CancerData into 70% training and 30% testing.
We can achieve above goal by following coding:

```{r}
library(caret) # load the caret package for knn
#use CancerData.csv 70% of dataset as training set and 30% as test set
indexes = createDataPartition(train_data$TARGET_deathRate, p = .7, list = F)
train = train_data[indexes, ]
test = train_data[-indexes, ]
```


## Develop KNN model for predicting Cancer Mortality. Evaluate test MSE for at least 5 different values of K
and find the K that minimizes test MSE.
I will develop KNN model by following coding (test MSE summary is in the end of this part):
```{r}
# Delete the "Geography" column
train_k = train[, !colnames(train) %in% "Geography"]
test_k = test[, !colnames(test) %in% "Geography"]

# Data preprocessing
train_x = train_k[, -21]
train_x = scale(train_x)[,]
train_y = train_k[,21]

test_x = test_k[, -21]
test_x = scale(test_k[,-21])[,]
test_y = test_k[,21]

# fit KNN model
knnmodel_1 = knnreg(train_x, train_y,k=1)
str(knnmodel_1)

# Calculate the MSE when k = 1
pred_y = predict(knnmodel_1, data.frame(test_x))
mse_1 = mean((test_y - pred_y)^2)
mse_1

knnmodel_1 = knnreg(train_x, train_y,k=1)
str(knnmodel_1)

# Calculate the MSE when k = 5
knnmodel_5 = knnreg(train_x, train_y,k=5)
pred_y = predict(knnmodel_5, data.frame(test_x))
mse_5 = mean((test_y - pred_y)^2)
mse_5

# Calculate the MSE when k = 10
knnmodel_10 = knnreg(train_x, train_y,k=10)
pred_y = predict(knnmodel_10, data.frame(test_x))
mse_10 = mean((test_y - pred_y)^2)
mse_10

# Calculate the MSE when k = 50
knnmodel_50 = knnreg(train_x, train_y,k=50)
pred_y = predict(knnmodel_50, data.frame(test_x))
mse_50 = mean((test_y - pred_y)^2)
mse_50

# Calculate the MSE when k = 100
knnmodel_100 = knnreg(train_x, train_y,k=100)
pred_y = predict(knnmodel_100, data.frame(test_x))
~~~~

```
mse_100 = mean((test_y - pred_y)^2)
mse_100
```

By varying the value of k in the KNN model, we can obtain different results for the test MSE. The test MSE values are as follows: k=1, MSE=16.8; k=5, MSE=11.5; k=10, MSE=11; k=50, MSE=13; and k=100, MSE=14.9. Based on this, it can be concluded that the KNN model has the lowest test MSE of 11 when k is equal to 10.

KNN is a non-linear technique, but does not work well with high dimensional data. Identify important variables from Linear Regression model and use only a subset of important fratures in the KNN model. And how it impacts on test performance.

From above linear regression analysis, we can know that the following variables are important: incidenceRate, medIncome, PctHS18_24, PctBachDeg18_24, PctMarriedHouseholds, povertyPercent, PctPublicCoverageAlone, PctOtherRace.

So, I will refit the knn model only by using those variables.

```{r}
# I need a new data set for important variables, so I will delete the unimportant variables.

train_re_k = train_k[, !colnames(train_k) %in%
c("MedianAge","MedianAgeMale","MedianAgeFemale","AvgHouseholdSize","PercentMarried","PctNoHS18_24","PctSomeCol1
","PctWhite","PctBlack","PctAsian")]
test_re_k = test_k[, !colnames(test_k) %in%
c("MedianAge","MedianAgeMale","MedianAgeFemale","AvgHouseholdSize","PercentMarried","PctNoHS18_24","PctSomeCol1
","PctWhite","PctBlack","PctAsian")]

# Data preprocessiong
train_rx = train_re_k[, -10]
train_rx = scale(train_rx)[,]
train_ry = train_re_k[,10]

test_rx = test_re_k[, -10]
test_rx = scale(test_re_k[,-10])[,]
test_ry = test_re_k[,10]

# fit KNN model where k = 10
knnmodel_r10 = knnreg(train_rx, train_ry,k=10)
str(knnmodel_r10)

# Calculate the MSE when k = 10
pred_ry = predict(knnmodel_r10, data.frame(test_rx))
mse_r10 = mean((test_ry - pred_ry)^2)
mse_r10
```

After selecting specific variables and using a fixed value of k=10, the test MSE increased compared to the previous results. There are several possible explanations for this observation.

Firstly, deleting unimportant variables can have resulted in the loss of crucial information necessary for accurate predictions, leading to an increase in the KNN MSE.

Secondly, removing variables can also have reduced the number of dimensions in the dataset, leading to the curse of dimensionality. This can cause computational requirements to exponentially increase as the number of dimensions increases, thereby reducing the KNN model's performance and increasing the MSE.

Lastly, the deleted variables can have been correlated with the important variables in the KNN model, and deleting them could have resulted in multicollinearity, leading to unstable and inaccurate predictions.

4.Feature Selection

“Executive Summary” section documenting interpretation of the important features impacting cancer mortality and how they influence cancer mortality.

The impact of cancer mortality is influenced by a multitude of factors, and several key features have been identified as important in determining mortality rates. From the results of the linear regression and KNN models, it is evident that several variables, including incidence rate, median household income, percentage of high school graduates aged 18-24, percentage of bachelor's degree holders aged 18-24, percentage of married households, poverty percentage, percentage of individuals with public coverage alone, and percentage of individuals belonging to other races, have a significant impact on cancer mortality.

Incidence rate, which represents the number of new cancer cases reported each year, is a critical determinant of cancer mortality. Higher incidence rates typically result in higher mortality rates, as more people are affected by the disease. Median household income is also a crucial factor, as individuals with higher incomes generally have better access to healthcare and are more likely to receive timely and effective cancer treatments.

The percentage of high school graduates and bachelor's degree holders aged 18-24 are also important variables, as education level is often associated with lifestyle factors that can impact cancer risk, such as smoking and diet. Additionally, individuals who are married and have a stable family life may have better access to healthcare and social support, which can positively impact their cancer outcomes.

Poverty percentage and the percentage of individuals with public coverage alone are also significant factors, as poverty can limit access to healthcare and effective cancer treatments. Finally, the percentage of individuals belonging to other races may also be an important factor, as some racial and ethnic groups may have a higher risk of certain types of cancer or may face barriers to accessing healthcare.

As a result, these features represent some of the most critical factors influencing cancer mortality rates, and addressing them could help to reduce cancer mortality rates and improve overall cancer outcomes.

5. Performance reporting on Holdout data
Summary and comparation about the model performance (MSE) of LR and KNN on holdout dataset as a table.

~~~
```{r}
# Using the linear regression model to predict the test data set
LR.pred = predict(lm.fit, newdata = test_data, type="response")
# Calculating the linear regression MSE
LR.mse_hold = mean((test_data$TARGET_deathRate - LR.pred)^2)
LR.mse_hold
```
```{r}
# Using the knn model to predict the test data set
# Test data preprocessing
K_test = test_data[, !colnames(test_data) %in%
c("Geography","MedianAge","MedianAgeMale","MedianAgeFemale","AvgHouseholdSize","PercentMarried","PctNoHS18_24",
","PctWhite","PctBlack","PctAsian")]

# Data preprocessing
test_x = K_test[, -10]
test_x = scale(K_test[,-10])[,]
test_y = K_test[,10]

# Using the fited KNN model where k = 10
knnmodel_r10 = knnreg(train_rx, train_ry,k=10)
str(knnmodel_r10)
# Calculate the MSE when k = 1
pred_y = predict(knnmodel_r10, data.frame(test_x))
Kmse_10 = mean((test_y - pred_y)^2)
Kmse_10

```
~~~

From the table above, we can see that the KNN model performed much better than the linear regression model in terms of test MSE when using the holdout dataset. The KNN model has a test MSE of 23.8, which is significantly lower than the linear regression model's test MSE of 407.7. This indicates that the KNN model is better suited for predicting cancer mortality rates based on the given features in the dataset.

In general, KNN models tend to perform better than linear regression models when dealing with non-linear relationships between the independent and dependent variables. This could be the case in this particular dataset, where there may be complex interactions and non-linear relationships between the features and the outcome variable. On the other hand, linear regression models assume a linear relationship between the variables, which may not be appropriate in all cases.

Overall, the KNN model's superior performance on the holdout dataset suggests that it may be a more appropriate model for predicting cancer mortality rates based on the given features in the dataset.