Data Analysis Notebook - MP2

Bella Cruz, Maria Larmon, Abhi Thanvi

2023-12-03

Contents

Our Team

Our team members and their roles were the following:

- **Bella Cruz:** Data Cleaning; Data Exploration; Model Fitting
- **Abhi Thanvi:** Formatting; Model Fitting; Predictive Selection and Power
- **Maria Larmon:** Classification tables; Predictive power; Cross-Validation

Data Exploration

```
#Reading our dataset
water_pot <- read.csv("data/water_potability.csv", header = TRUE)

head(water_pot)
```

```
##         ph Hardness   Solids Chloramines  Sulfate Conductivity Organic_carbon
## 1       NA 204.8905 20791.32    7.300212 368.5164     564.3087      10.379783
## 2 3.716080 129.4229 18630.06    6.635246       NA     592.8854      15.180013
## 3 8.099124 224.2363 19909.54    9.275884       NA     418.6062      16.868637
## 4 8.316766 214.3734 22018.42    8.059332 356.8861     363.2665      18.436524
## 5 9.092223 181.1015 17978.99    6.546600 310.1357     398.4108      11.558279
## 6 5.584087 188.3133 28748.69    7.544869 326.6784     280.4679       8.399735
##   Trihalomethanes Turbidity Potability
## 1        86.99097  2.963135          0
## 2        56.32908  4.500656          0
## 3        66.42009  3.055934          0
## 4       100.34167  4.628771          0
## 5        31.99799  4.075075          0
## 6        54.91786  2.559708          0
```

The dataset initially has 3276 rows and 10 columns.

Columns:

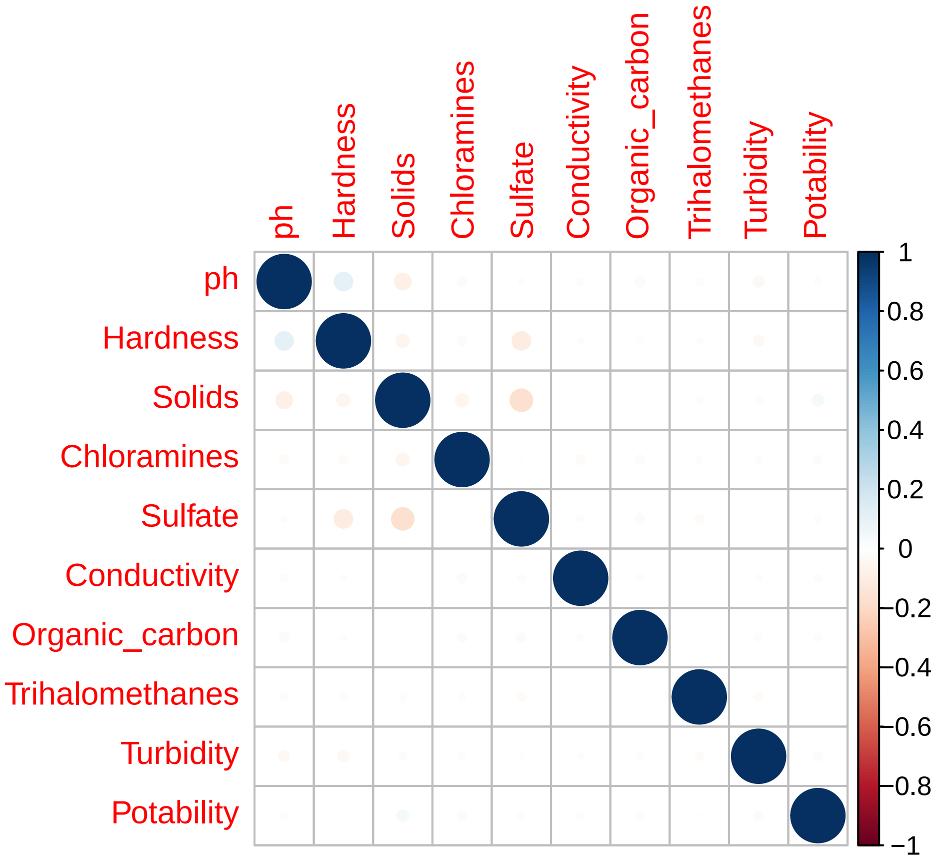
- **ph:** Represents the pH levels.
- **Hardness:** Indicates water hardness.
- **Solids:** Denotes the concentration of dissolved solids.
- **Chloramines:** Reflects the presence of chloramines in the water.
- **Sulfate:** Indicates sulfate levels.
- **Conductivity:** Represents the electrical conductivity of the water.
- **Organic_carbon:** Denotes the concentration of organic carbon.
- **Trihalomethanes:** Reflects the presence of trihalomethanes.
- **Turbidity:** Indicates water turbidity.
- **Potability:** Binary variable (0 or 1) indicating water potability, where 1 represents potable water.

Please note that some values are missing (NA) in the dataset so we drop them before furthering our analysis. Now our row count is 2011 and we call our data as `df` for simplicity.

Part A: Data Exploration

This section is a exploratory analysis of this data set in order to evaluate the water quality attributes and their relationship with the drinking water status. We aim to better understand what patterns we are dealing with in our data to make a better decision during modelling process.

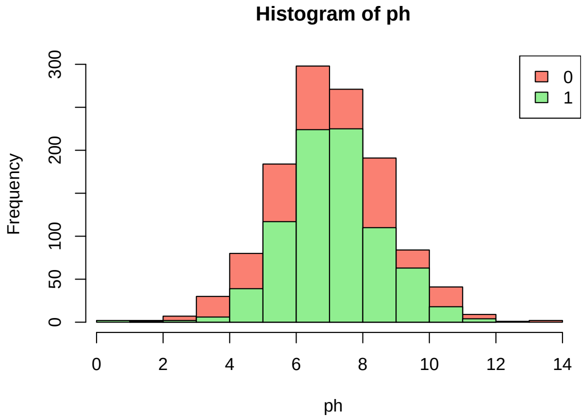
```
corrplot(cor(df))
```

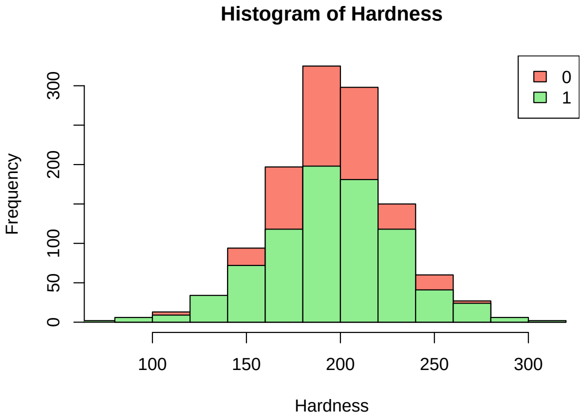


Based on the correlation matrix plot, it seems like the dataset is forgiving in that no two variables are strongly correlated with one another. So, we should be good and not see any multicollinearity issue when fitting model.

BUT when we initially fit a full linear logit model (can be seen in our Analysis.Rmd) on the data, we saw an issue that none of our predictors were significant. So we decided to look more closely at how our data is dispersed.

Dispersion of Data





The `pH vs. Potability` graph and `Hardness vs. Potability` histogram graphs both imply that the dispersion of points for when the water is not potable (Potability=0) tends to cluster around the center range of their respective levels compared to the dispersion of points when the water is deemed potable (Potability=1). This could have been the reason for our main effects model to not suffice and suggest that we consider adding quadratic terms into the fitted model as well.

Part B. Model Fitting

This section fits an appropriate model to this data set in order to make predictions about the potability of water given a set of measurements on water quality attributes. We learnt that a Main Effects model will not suffice and therefore will attempt to fit a model with quadratic terms and see the results.

```r
full_quad_mod <- glm(Potability ~
                       ph + I(ph^2) +
                       Hardness + I(Hardness^2) +
                       Solids + I(Solids^2) +
                       Chloramines + I(Chloramines^2) +
                       Sulfate + I(Sulfate^2) +
                       Conductivity + I(Conductivity^2) +
                       Organic_carbon + I(Organic_carbon^2) +
                       Trihalomethanes + I(Trihalomethanes^2) +
                       Turbidity + I(Turbidity^2), family=binomial, data=df)
summary(full_quad_mod)
```

```
##
## Call:
## glm(formula = Potability ~ ph + I(ph^2) + Hardness + I(Hardness^2) +
##     Solids + I(Solids^2) + Chloramines + I(Chloramines^2) + Sulfate +
##     I(Sulfate^2) + Conductivity + I(Conductivity^2) + Organic_carbon +
##     I(Organic_carbon^2) + Trihalomethanes + I(Trihalomethanes^2) +
##     Turbidity + I(Turbidity^2), family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9775  -0.9851  -0.8652   1.2885   2.4260
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.415e+01  2.894e+00   4.890 1.01e-06 ***
## ph                    9.563e-01  2.080e-01   4.599 4.25e-06 ***
## I(ph^2)              -6.544e-02  1.439e-02  -4.549 5.40e-06 ***
## Hardness             -3.656e-02  1.129e-02  -3.240  0.00120 **
## I(Hardness^2)         9.306e-05  2.868e-05   3.244  0.00118 **
## Solids                8.707e-06  2.347e-05   0.371  0.71060
## I(Solids^2)           1.479e-11  4.674e-10   0.032  0.97476
## Chloramines          -5.348e-01  1.782e-01  -3.000  0.00270 **
## I(Chloramines^2)      3.946e-02  1.237e-02   3.190  0.00142 **
## Sulfate              -7.494e-02  1.345e-02  -5.570 2.55e-08 ***
## I(Sulfate^2)          1.119e-04  2.001e-05   5.592 2.24e-08 ***
## Conductivity         -5.466e-03  4.823e-03  -1.133  0.25708
## I(Conductivity^2)     5.716e-06  5.473e-06   1.044  0.29627
## Organic_carbon        1.032e-01  8.751e-02   1.179  0.23844
## I(Organic_carbon^2)  -3.747e-03  3.016e-03  -1.242  0.21414
## Trihalomethanes      -1.212e-02  1.613e-02  -0.752  0.45227
## I(Trihalomethanes^2)  9.967e-05  1.201e-04   0.830  0.40653
## Turbidity             1.578e-01  4.408e-01   0.358  0.72027
## I(Turbidity^2)       -1.352e-02  5.508e-02  -0.245  0.80616
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2712.1  on 2010  degrees of freedom
## Residual deviance: 2609.3  on 1992  degrees of freedom
## AIC: 2647.3
##
## Number of Fisher Scoring iterations: 4
```

Perfect! Now at least we can see some predictors being significant. This means this transformation/switch was good. We will later do predictor selection so we can move on from this section!

Part C. Predictor Selection

We just chose a full model with quadratic terms (`full_quad_model`) and saw better results. However, only few predictors were significant while others were not. In this section, we aim to select the best predictors determining the water potability.

We will consider all three `forward, backward, step-wise` selection methods and choose the one that produces model (i.e. smallest AIC)

Selection Algorithms

```
intercept_model <- glm(Potability ~ 1, family=binomial, data=df)

# Forward Selection
forward_mod <- step(intercept_model, ~ ph + I(ph^2) + Hardness + I(Hardness^2)
                    + Solids + I(Solids^2) + Chloramines + I(Chloramines^2)
                    + Sulfate + I(Sulfate^2) + Conductivity + I(Conductivity^2)
                    + Organic_carbon + I(Organic_carbon^2)
                    + Trihalomethanes + I(Trihalomethanes^2)
                    + Turbidity + I(Turbidity^2), direction="forward", trace=0)
summary(forward_mod)
```

```
##
## Call:
## glm(formula = Potability ~ I(Solids^2) + I(Chloramines^2) + Chloramines,
##     family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6952  -0.9963  -0.9500   1.3463   1.4518
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.732e+00  6.057e-01   2.860  0.00424 **
## I(Solids^2)       1.930e-10  1.041e-10   1.854  0.06379 .
## I(Chloramines^2)  5.041e-02  1.169e-02   4.311 1.63e-05 ***
## Chloramines      -6.905e-01  1.689e-01  -4.088 4.35e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2712.1  on 2010  degrees of freedom
## Residual deviance: 2687.8  on 2007  degrees of freedom
## AIC: 2695.8
##
## Number of Fisher Scoring iterations: 4
```

```
# Backward Selection
backward_mod <- step(full_quad_mod, direction="backward", trace=0)
summary(backward_mod)
```

```
##
## Call:
## glm(formula = Potability ~ ph + I(ph^2) + Hardness + I(Hardness^2) +
##     Solids + Chloramines + I(Chloramines^2) + Sulfate + I(Sulfate^2),
##     family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9675  -0.9825  -0.8790   1.2988   2.4330
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.353e+01  2.384e+00   5.676 1.38e-08 ***
## ph                9.712e-01  2.071e-01   4.690 2.73e-06 ***
## I(ph^2)          -6.648e-02  1.432e-02  -4.644 3.42e-06 ***
## Hardness         -3.623e-02  1.127e-02  -3.216  0.00130 **
## I(Hardness^2)     9.195e-05  2.862e-05   3.213  0.00132 **
## Solids            9.031e-06  5.542e-06   1.629  0.10322
## Chloramines      -5.172e-01  1.771e-01  -2.920  0.00350 **
## I(Chloramines^2)  3.822e-02  1.229e-02   3.109  0.00188 **
## Sulfate          -7.518e-02  1.336e-02  -5.627 1.84e-08 ***
## I(Sulfate^2)      1.123e-04  1.987e-05   5.653 1.58e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2712.1  on 2010  degrees of freedom
## Residual deviance: 2614.3  on 2001  degrees of freedom
## AIC: 2634.3
##
## Number of Fisher Scoring iterations: 4
```

```r
# Step Selection
step_mod <- step(intercept_model, ~ ph + I(ph^2) + Hardness + I(Hardness^2)
                    + Solids + I(Solids^2) + Chloramines + I(Chloramines^2)
                    + Sulfate + I(Sulfate^2) + Conductivity + I(Conductivity^2)
                    + Organic_carbon + I(Organic_carbon^2)
                    + Trihalomethanes + I(Trihalomethanes^2)
                    + Turbidity + I(Turbidity^2), direction="both", trace=0)
summary(step_mod)
```

```
##
## Call:
## glm(formula = Potability ~ I(Solids^2) + I(Chloramines^2) + Chloramines,
##     family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6952  -0.9963  -0.9500   1.3463   1.4518
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.732e+00  6.057e-01   2.860  0.00424 **
## I(Solids^2)       1.930e-10  1.041e-10   1.854  0.06379 .
## I(Chloramines^2)  5.041e-02  1.169e-02   4.311 1.63e-05 ***
## Chloramines      -6.905e-01  1.689e-01  -4.088 4.35e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2712.1  on 2010  degrees of freedom
## Residual deviance: 2687.8  on 2007  degrees of freedom
## AIC: 2695.8
##
## Number of Fisher Scoring iterations: 4
```

Observations:

- Seems like `Forward Selection` and `Step-Wise Selection` both converged to the same subset of predictors
  - `Intercept`, `Solids^2^` (not super signif.), `Chloramines^2^`, `Chloromines`
  - Both have **AIC = 2695.8**
- `Backward Selection` had a longer list of selected predictors and can be seen above
  - `Solids` is selected in the model but is not significant (we shall see what to with this later)
  - Has the smaller **AIC = 2634.3** Since the `Backward Selection` produced the smaller AIC among the three selection algorithms, we shall move forward with running diagnositcs on `backward_mod`.

Model Diagnostics - Backward Selected

NEED TO DECIDE IF I NEED TO REMOVE ‘Solids‘ as predictor from the model....OH Monday to Confirm...

```
drop_results <- drop1(backward_mod, test = "Chisq")

# Create a data frame with variable names and their p-values
drop_terms <- data.frame(
  Variable = rownames(drop_results),
  P_Value = drop_results$Pr
)

# Print the terms and their significance
print(drop_terms)
```

```
##            Variable      P_Value
## 1            <none>           NA
## 2                ph 3.645138e-07
## 3           I(ph^2) 4.886640e-07
## 4          Hardness 1.250932e-03
## 5     I(Hardness^2) 1.265726e-03
## 6            Solids 1.033434e-01
## 7       Chloramines 3.366765e-03
## 8  I(Chloramines^2) 1.776087e-03
## 9           Sulfate 2.247767e-09
## 10     I(Sulfate^2) 2.010983e-09
```

```r
# Remove 'Solids' if it's in the list of drop-able terms
if ('Solids' %in% drop_terms$Variable[drop_terms$P_Value > 0.05]) {
  tmp_model <- update(backward_mod, . ~ . - Solids)
} else {
  tmp_model <- backward_mod
}

# Display the summary of the final model
summary(tmp_model)
```

```
##
## Call:
## glm(formula = Potability ~ ph + I(ph^2) + Hardness + I(Hardness^2) +
##     Chloramines + I(Chloramines^2) + Sulfate + I(Sulfate^2),
##     family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9398  -0.9798  -0.8802   1.3060   2.4592
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.404e+01  2.362e+00   5.946 2.75e-09 ***
## ph                9.569e-01  2.067e-01   4.630 3.65e-06 ***
## I(ph^2)          -6.573e-02  1.429e-02  -4.601 4.21e-06 ***
## Hardness         -3.646e-02  1.125e-02  -3.240  0.00120 **
## I(Hardness^2)     9.218e-05  2.859e-05   3.224  0.00126 **
## Chloramines      -5.175e-01  1.768e-01  -2.927  0.00342 **
## I(Chloramines^2)  3.808e-02  1.227e-02   3.103  0.00191 **
## Sulfate          -7.614e-02  1.335e-02  -5.704 1.17e-08 ***
## I(Sulfate^2)      1.133e-04  1.986e-05   5.706 1.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2712.1  on 2010  degrees of freedom
## Residual deviance: 2617.0  on 2002  degrees of freedom
## AIC: 2635
##
## Number of Fisher Scoring iterations: 4
```

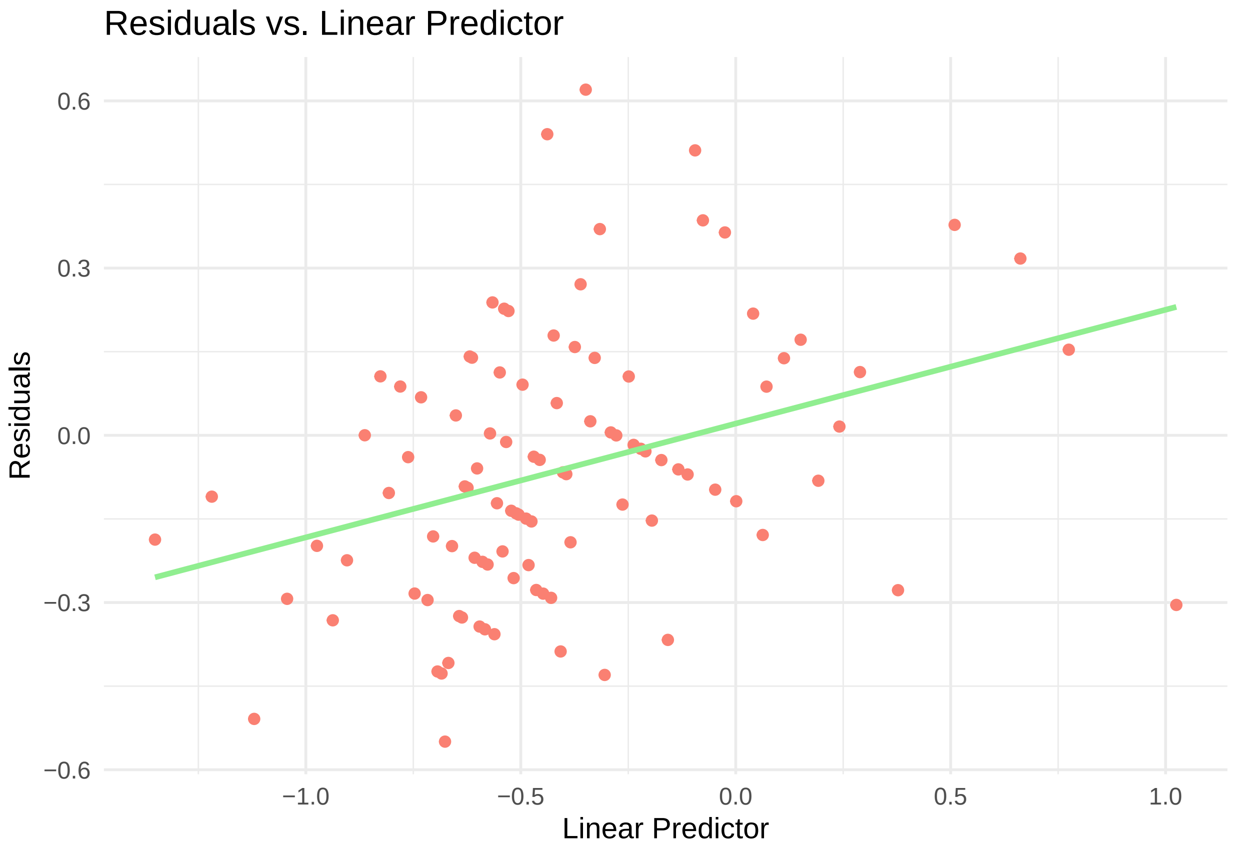
Dropping `Solids` Increased our AIC by 0.7. We are at a fork here. But we will use the Backward Selection model **with** the `Solids` predictor for 2 reasons:

- Most importantly, the AIC increased slightly by removing the predictor
- Secondly, in real world, it would make sense that solids dissolved in the water would be one of important factor to determine potability.

```r
library(dplyr)

tmp_df <- df %>%
  mutate(
    residuals = residuals(backward_mod),
    linpred = predict(backward_mod)
  ) %>%
  group_by(bin = cut(linpred, breaks = unique(quantile(linpred, (1:100)/101)))) %>%
  summarise(
    residuals = mean(residuals),
    linpred = mean(linpred)
  )

# Create a smoother line plot using ggplot2
ggplot(tmp_df, aes(x = linpred, y = residuals, group = 1)) +
  geom_point(color = "salmon") +
  geom_smooth(method = "lm", se = FALSE, color = "lightgreen") +
  labs(
    title = "Residuals vs. Linear Predictor",
    x = "Linear Predictor",
    y = "Residuals"
  ) +
  theme_minimal()
```



The residual plots looks okay because the residuals are following the general trend of the linear predictor and seems like are equally below and above the prediction line. However, we need to run more diagnositcs since our data is not grouped which makes us unconfident in our diagnostic-analysis.

```
# ....NEED TO RUN MORE DIAGNOSTICS LATER....
```

Model Evaluation - Predictive Power

This section summarizes the predictive power of the final model selected, using correlation based measures and likelihood based measures.

```
# correlation measure final model
cor(df$Potability, fitted(backward_mod))
```

```
## [1] 0.2195577
```

A correlation coefficient of 0.22 suggests a positive but weak linear relationship between the observed `Potability` values and the fitted values from the logistic regression model we got from backward selection.

```
# likelihood measure final model
lik <- (logLik(backward_mod) - logLik(intercept_model))/(0-logLik(intercept_model))
as.numeric(lik)
```

```
## [1] 0.0360592
```

Although, `Backward Model` gave us the lowest AIC, the Likelihood Test Statistic is 0.036 which is relatively small. Indicating that our `Backward Model` is not significantly better at prediction compared to the `Intercept Model`.

Model Evaluation - Classification Tables

This section uses classification tables to measure the predictive power of the best final model selected based on the results obtained in (c). Implement the leave-one-out cross-validation method for this step.

```
pi0 <- 0.5
class_table = table(y=df$Potability, yhat=as.numeric(fitted(backward_mod) > pi0))
class_table
```

```
##    yhat
## y      0    1
##   0 1093  107
##   1  649  162
```

```
sensitivity(class_table)
```

```
## # A tibble: 1 x 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 sensitivity binary         0.627
```

```
specificity(class_table)
```

```
## # A tibble: 1 x 3
##   .metric     .estimator .estimate
##   <chr>       <chr>          <dbl>
## 1 specificity binary         0.602
```

```
posPredValue(class_table)
```

```
## [1] 0.9108333
```

```
accuracy(class_table)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.624
```

Using the backwards model we are able to create a classification table and then calculate more information about the predictive power of the final model. For our final model: our sensitivity is 0.627, our specificity is 0.602, The overall accuracy of our model is about 62.4% and the positive predictive value of our model is about 0.91

```
df_fact = df |>
  mutate(Potability = factor(Potability))
```

```r
controlLoo <- trainControl(method="LOOCV")

loocv_model <- train(Potability ~ ph + I(ph^2) + Hardness + I(Hardness^2) + Solids +
        Chloramines + I(Chloramines^2) + Sulfate + I(Sulfate^2), data=df_fact,
        trControl=controlLoo, method = "glm")

print(loocv_model)
```

```
## Generalized Linear Model
##
## 2011 samples
##    5 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 2010, 2010, 2010, 2010, 2010, 2010, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6185977  0.1114743
```

Using the leave-one-out cross-validation method to ensure that our data is not overly- optimistic, we receive an overall accuracy of about 61.9%. This may be a better estimation of our model's actual accuracy because it cross validates against the data.

Note: Note: You can use the heart disease dataset analysis posted in Canvas Module 6 as a guide for your analysis. Please note that this is not an extensive analysis but you can use the pieces of code that you need. More details about this example can be found in Faraway, J.J. (2016): Extending the Linear Model with R, Chapter 2. A good reference about model diagnostics can be found in Dunn, P/K. and Smyth, G.K. (2018): Generalized Linear Models with Examples in R, Chapter 8.