Prediction Models for Marine Water Quality in Hong Kong

Prediction and monitoring of beach water quality can notify the public about potential hazardous levels of contamination and is therefore an important measure of public safety. A key indicator for beach water quality is the bacterial level of E. coli, due to its correlation with swimming relating illnesses (Chan et al, 2013). The bacteria itself can cause illness with minor consequences, but more importantly acts as an indicator for other pathogens (Thoe et al, 2012). Measurement of E. coli levels are a time-consuming process which takes at least 24 hours. As E. coli levels changes frequently due to hydro-meteorological factors, conventional measurement does not serve as an adequate way of informing the public about hazardous levels of the bacteria (Thoe & Lee, 2014). Therefore, predicting E. coli levels with statistical models have been addressed as a solution to warn about contamination. Over the past few years, prediction models to estimate water quality have emerged in research various locations, in the United States (Thoe et al, 2014), Hong Kong (Thoe & Lee, 2014) and Pakistan (Ahmed et al, 2019) to name a few. In Hong Kong, 41 multilinear regression models have been made for each beach as an advisory system (Thoe & Lee, 2014). The variables used to predict E. coli in these systems are based on climatological data as well as previously recorded E. coli levels and salinity. However, no other water quality parameters were used for the prediction. Other prediction models have included the use of additional water quality parameters, including pH, oxygen levels and other dissolved solids (Ahmed et al, 2019). As there is public data on a wide range of water quality parameters in Hong Kong, the goal of this project is to predict E. coli levels at the city beaches by exploring a broad range of variables and find the best fitting model.

Climatological and water quality data was used for the prediction model of E. coli levels. The dataset for water quality data was collected at a Hong Kong governmental website for open data (DATA, 2020), while the climatological datasets were collected from the website of Hong Kong Observatory (*Climatological Information Services, 2020*). To have an adequate number of observations in the datasets used for prediction and testing, data from 2004 to 2018 was collected. The water quality data was given in datasets covering a year, and each yearly dataset contained all the relevant parameters. Also, the observations in these datasets were classified into “water zones” where the measurements occurred. The climatological data was also given in datasets covering a year, but only contained one parameter per dataset. Therefore, a more extensive data collection process was needed to acquire all the relevant climatological data. The acquired datasets were imported into R, cleaned for missing values, and merged by the shared “Date” variable. Furthermore, several of the water quality parameters (including E. coli levels which was the response in the prediction) were transformed to their natural logarithmic value. These variables had an exponential distribution, which lead to poorer prediction in comparison to their natural logarithm. The transformed, logarithmic variables were closer to a normal distribution, causing the final dataset to be more suitable for prediction. Also, several variables with past day’s measurement values were added to each observation. These variables were added based on past research (Thoe & Lee, 2014) in E. coli prediction, where the variables served as significant predictors. More specifically, these variables were the past day E. coli measurement, past day solar radiation, past day wind direction and the past day wind speed. The values of these variables were the average of the past day’s measurements. Unfortunately, the values could not be averaged over the specific locations (“water zones”) as there were no daily observations from one specific location in the data set.

For prediction of the E. coli response variable, several multilinear regression methods and regression trees were compared by their accuracy in the form of the R2­ coefficient­­. The linear regression methods included least squares, least squares with principal component analysis, best subsets, lasso, and ridge regression. For the tree regressions, the methods used were regression tree analysis, random forest, bagged tree and boosted tree. The number of principle components (least squares) and the penalty term (for lasso and ridge regression), cost complexity (decision tree and bagged tree), mtry (random forest) and learning rate (boosted tree) were tuned by using ten-fold cross-validation, and the tuned parameter that gave the highest R2 was used.

|  |  |
| --- | --- |
| **Model** | **Test R2** |
| Best subset regression | 0.491 |
| Lasso regression | 0.474 |
| Ridge regression | 0.477 |
| Decision tree regression | 0.311 |
| Bagged tree regression | 0.414 |
| Boosted tree regression | 0,421 |
| Random forest regression | 0,453 |

Table 1: Test R2 for each regression model

|  |
| --- |
| **Coefficients chosen by best subset regression** |
| Biochemical oxygen demand (natural logarithm) |
| Ammonia nitrogen (natural logarithm) |
| Dissolved oxygen (natural logarithm) |
| Nitrate (natural logarithm) |
| Orthophosphate phosphorus (natural logarithm) |
| Silica (natural logarithm) |
| Sea temperature |
| Max temperature |
| Past day E. coli average (natural logarithm) |

Table 2: coefficients chosen by best subset regression. The variables that were transformed to their logarithmic are commented on in brackets.

The model which gave the highest adjusted R2 value (0.491) on the testing set was best subset regression, while the highest R2 value (0.7) overall occurred in one of the cross-validation folds during the tuning of the boosted tree regression (0.7). The significant difference in training error (0.7) and testing error (0.421) achieved by the boosted tree model indicates that too much variance was picked up by the model in the training set, causing the model to overfit the testing set. Therefore, the best subset regression achieved the highest adjusted R2 value, making the model most suitable to predict E. coli levels with the given input data. Furthermore, this R2 value is on the upper boundary of the R2 value found in previous research regarding E. coli prediction, as the R2 value typically ranges from 30-50% (Thoe & Lee, 2014; Thoe et al, 2014; Francy et al, 2006; Thoe et al, 2012). Therefore, one can conclude that the model meets general prediction standards. As one can see in table 2, the optimal combination of coefficients from the best subsets model are mainly water quality parameters obtained from the governmental water quality dataset, except for the parameters “sea temperature” and “max temperature”, which both originate from the climatological data. This result reflects the general trend of poor correlation between the climatological data and E. coli levels in the final dataset used for analysis (ranged from -0.135 to 0.063). A possible explanation for this poor correlation might be due to the large number of locations used in the regression analysis, as all locations in Hong Kong where the E. coli measurements took place are included. E. coli levels changes rapidly due to environmental factors at specific locations, like beaches (Thoe & et al, 2014), meaning climatological factors might have a local impact. However, such local impact on the E. coli levels will not necessarily be accounted for in a city-scale model. This may be the reason why the regression model had low correlation with climatological variables in comparison to other models that aimed to predict E. coli levels at specific beaches (Thoe & Lee, 2014; Thoe et al, 2014; Francy et al, 2006).

Despite the relatively high R2 value achieved by the prediction, there are several drawbacks by this model and room for improvement. First, the model predicts E. coli levels at a city-wide scale, which may overlook local conditions at beaches. Local conditions can cause local variation in E. coli levels, and such variation could possibly be underestimated by an advisory system if the prediction model covers to large of an area. For the model to be more location specific, more observations would be needed more frequently from each location in the data set used for analysis. In this way, the variables containing past-day measurements in each observation can originate from the same location (“water zone”) as the rest of the observation. Currently, the variables with past day measurements are averages of the available data from the day before (which does not originate from the same location). Furthermore, the model is heavily based on water quality parameters (seen in table 1) for prediction. Consequently, a daily supply of these water quality parameters is needed as input to undergo the prediction. The extensiveness of measuring these parameters has not been investigated in this project, meaning there is further room for understanding the efficiency of this model.

To conclude, a prediction model was created to forecast E. coli levels in the waters of Hong Kong. As E. coli levels change rapidly to environmental factors and measurements are time demanding, there is a need to predict such levels in order prevent public exposure to possibly dangerous levels of contamination. The datasets used in the analysis were acquired from a governmental website (DATA, 2020) and Hong Kong Observation (*Climatological Information Services, 2020*). For the regression analysis, various linear regressions and tree regressions were done, and their resulting test predictions were compared. The highest R2 value (0.491) was acquired with a best subsets regression model, which is a prediction result in the upper boundary (30-50%) of normal R2 values for E. coli prediction. The prediction model used mostly water quality parameters, while the climatological data appeared to have low correlation with the response. This low correlation might be because the model’s prediction for E. coli levels occurs over the entire city of Hong Kong, while the climatological data has more of a local impact the E. coli levels, which is then not accounted for in the model. If this is the case, the model could be improved by focusing more specifically on beaches instead of the whole city. However, this would first demand a more extensive collection of data, which would be needed to be done on a close-to daily rate for each location. Also, the efficiency of the model could be further explored by investigating how demanding it is to collect the water quality parameters daily.

References

Chan, S. N., Thoe, W., & Lee, J. H. W. (2013). Real-time forecasting of Hong Kong beach water quality by 3D deterministic model. *Water Research*, *47*(4), 1631–1647. <https://doi.org/10.1016/j.watres.2012.12.026>

Thoe, W., Wong, S. H. C., Choi, K. W., & Lee, J. H. W. (2012). Daily prediction of marine beach water quality in Hong Kong. *Journal of Hydro-Environment Research*, *6*(3), 164–180. <https://doi.org/10.1016/j.jher.2012.05.003>

Thoe, W., & Lee, J. H. W. (2014). Daily Forecasting of Hong Kong Beach Water Quality by Multiple Linear Regression Models. *Journal of Environmental Engineering*, *140*(2),. <https://doi.org/10.1061/(ASCE)EE.1943-7870.0000800>

Thoe, W., Gold, M., Griesbach, A., Grimmer, M., Taggart, M. L., & Boehm, A. B. (2014). Predicting water quality at Santa Monica Beach: Evaluation of five different models for public notification of unsafe swimming conditions. *Water Research*, *67*, 105–117. <https://doi.org/10.1016/j.watres.2014.09.001>

Ahmed, A. N., Othman, F. B., Afan, H. A., Ibrahim, R. K., Fai, C. M., Hossain, M. S., Ehteram, M., & Elshafie, A. (2019). Machine learning methods for better water quality prediction. *Journal of Hydrology*, *578*, 1. <https://doi.org/10.1016/j.jhydrol.2019.124084>

Francy, D. S., Darner, R. A., Bertke, E. E., Ohio Water Development Authority, Ohio Lake Erie Office, Northeast Ohio Regional Sewer District, & Cuyahoga County Board of Health. (2006). *Models for Predicting Recreational Water Quality at Lake Erie Beaches* (No. 2006–5192). USGS. <https://pubs.usgs.gov/sir/2006/5192/pdf/SIR2006-5192.pdf>

*DATA*. (2020). DATA.GOV.HK. <https://data.gov.hk/en-datasets>

*Climatological Information Services*. (2020). Hong Kong Observatory. <https://www.hko.gov.hk/en/cis/climat.htm>

Appendix A

R code:

library(readxl)

library(GGally)

library(ggfortify)

library(ISLR)

library(MASS)

library(tidyverse)

library(kknn)

library(discrim)

library(klaR)

library(tidymodels)

library(leaps)

library(gam)

library(rpart)

library(baguette)

library(xgboost)

library(lubridate)

library(rsq)

# Work directory

setwd("C:/Users/Bruker/Documents/University stuff/Semester 3/Machine Learning/Project")

# Functions

# for uploading and structuring the climatological data

upload <- function(folder, parameter, year){

table <- read\_excel(paste0(folder, parameter," ", year, ".xlsx")) %>% mutate(Year = year)

table <- table %>% rename("Okt" = Oct, "Mai" = May, "Des" = Dec)

table <- table %>% pivot\_longer(cols = !c(Day, Year), names\_to = "Month", values\_to = "col\_name")

table <- table %>% unite(Dates, Year, Month, Day, sep="/") %>% mutate(Dates = as.Date(Dates, "%Y/%b/%d"))

table <- table %>% mutate(Dates = as.Date(Dates, "%Y/%b/%d"))

}

# Datasets

# water quality

files <- list.files(path = "WQ data", pattern="\*.csv")

WaterQ <- list()

for (i in 1:length(files)) {

WaterQ[[i]] <- read\_csv(paste0(getwd(),"/WQ data/", files[i]))

}

WaterQ <- map(WaterQ, as\_tibble) %>%

bind\_rows() %>%

transmute(

Water\_zone = Water\_zone,

Station = Station,

Sample\_no = Sample\_no,

Depth = Depth,

Dates = as.Date(Dates),

# Replacing the E. Coli variable with its logarithmic value

lnEcoli = log(E\_coli),

lnBiochem\_ox\_demand = log(`5-day\_biochemical\_oxygen\_demand`),

lnAmmonia\_nitrogen = log(Ammonia\_nitrogen),

lnChlorophyll\_a = log(`Chlorophyll-a`),

lnDissolved\_ox = log(dissolved\_oxygen),

lnDissolved\_ox1 = log(dissolved\_oxygen\_1),

lnNitrate = log(Nitrate),

lnNitrite = log(Nitrite),

lnOrthophosphate\_phosphorus = log(Orthophosphate\_phosphorus),

lnPhaeo\_pigments = log(`Phaeo-pigments`),

lnSilica = log(Silica),

lnSuspended\_solids = log(Suspended\_solids),

lnTot\_inorganic\_nitrogen = log(Total\_inorganic\_nitrogen),

lnKjeldahl\_nitrogen = log(Kjeldahl\_nitrogen),

lnNitrogen = log(Nitrogen),

lnPhosphorus = log(Phosphorus),

lnTurbidity = log(Turbidity),

lnUnionised\_ammonia = log(Unionised\_ammonia),

lnVolatile\_suspended\_solids = log(Volatile\_suspended\_solids),

Temperature = Temperature,

pH = pH,

) %>%

filter(Depth == "Surface Water")

# climatological data

# daily wind speed

wind\_speed\_folder <- paste0(getwd(), "/Daily mean wind speed (km per h) - Wagland Island/")

wind\_speed <- list()

wind\_speed[[1]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2004")

wind\_speed[[2]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2005")

wind\_speed[[3]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2006")

wind\_speed[[4]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2007")

wind\_speed[[5]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2008")

wind\_speed[[6]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2009")

wind\_speed[[7]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2010")

wind\_speed[[8]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2011")

wind\_speed[[9]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2012")

wind\_speed[[10]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2013")

wind\_speed[[11]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2014")

wind\_speed[[12]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2015")

wind\_speed[[13]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2016")

wind\_speed[[14]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2017")

wind\_speed[[15]] <- upload(wind\_speed\_folder, "Daily mean wind speed", "2018")

wind\_speed <- map(wind\_speed, as\_tibble) %>% bind\_rows() %>% rename("wind\_speed" = col\_name)

# daily wind direction

wind\_direction\_folder <- paste0(getwd(), "/Daily prevailing wind direction (degrees) - Waglan Island/")

wind\_direction <- list()

wind\_direction[[1]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2004")

wind\_direction[[2]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2005")

wind\_direction[[3]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2006")

wind\_direction[[4]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2007")

wind\_direction[[5]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2008")

wind\_direction[[6]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2009")

wind\_direction[[7]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2010")

wind\_direction[[8]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2011")

wind\_direction[[9]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2012")

wind\_direction[[10]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2013")

wind\_direction[[11]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2014")

wind\_direction[[12]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2015")

wind\_direction[[13]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2016")

wind\_direction[[14]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2017")

wind\_direction[[15]] <- upload(wind\_direction\_folder, "Daily prevailing wind direction", "2018")

wind\_direction <- map(wind\_direction, as\_tibble) %>%

bind\_rows() %>%

rename("wind\_direction" = col\_name) %>%

transmute(Dates = Dates, lnWind\_direction = log(wind\_direction))

# daily rainfall

rainfall\_folder <- paste0(getwd(), "/Daily rainfall - HKO/")

rainfall <- list()

rainfall[[1]] <- upload(rainfall\_folder, "Daily rainfall", "2004")

rainfall[[2]] <- upload(rainfall\_folder, "Daily rainfall", "2005")

rainfall[[3]] <- upload(rainfall\_folder, "Daily rainfall", "2006")

rainfall[[4]] <- upload(rainfall\_folder, "Daily rainfall", "2007")

rainfall[[5]] <- upload(rainfall\_folder, "Daily rainfall", "2008")

rainfall[[6]] <- upload(rainfall\_folder, "Daily rainfall", "2009")

rainfall[[7]] <- upload(rainfall\_folder, "Daily rainfall", "2010")

rainfall[[8]] <- upload(rainfall\_folder, "Daily rainfall", "2011")

rainfall[[9]] <- upload(rainfall\_folder, "Daily rainfall", "2012")

rainfall[[10]] <- upload(rainfall\_folder, "Daily rainfall", "2013")

rainfall[[11]] <- upload(rainfall\_folder, "Daily rainfall", "2014")

rainfall[[12]] <- upload(rainfall\_folder, "Daily rainfall", "2015")

rainfall[[13]] <- upload(rainfall\_folder, "Daily rainfall", "2016")

rainfall[[14]] <- upload(rainfall\_folder, "Daily rainfall", "2017")

rainfall[[15]] <- upload(rainfall\_folder, "Daily rainfall", "2018")

rainfall <- map(rainfall, as\_tibble) %>% bind\_rows() %>% rename("rainfall" = col\_name)

# sea temp

sea\_temp\_folder <- paste0(getwd(), "/Daily sea temp - North Point/")

sea\_temp <- list()

sea\_temp[[1]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2004")

sea\_temp[[2]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2005")

sea\_temp[[3]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2006")

sea\_temp[[4]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2007")

sea\_temp[[5]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2008")

sea\_temp[[6]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2009")

sea\_temp[[7]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2010")

sea\_temp[[8]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2011")

sea\_temp[[9]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2012")

sea\_temp[[10]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2013")

sea\_temp[[11]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2014")

sea\_temp[[12]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2015")

sea\_temp[[13]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2016")

sea\_temp[[14]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2017")

sea\_temp[[15]] <- upload(sea\_temp\_folder, "Daily mean sea temperature", "2018")

sea\_temp <- map(sea\_temp, as\_tibble) %>% bind\_rows() %>% rename("sea\_temperature" = col\_name)

# solar radiation

solar\_radiation\_folder <- paste0(getwd(), "/Daily solar radiation - Kings Park/")

solar\_radiation <- list()

solar\_radiation[[1]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2004")

solar\_radiation[[2]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2005")

solar\_radiation[[3]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2006")

solar\_radiation[[4]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2007")

solar\_radiation[[5]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2008")

solar\_radiation[[6]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2009")

solar\_radiation[[7]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2010")

solar\_radiation[[8]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2011")

solar\_radiation[[9]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2012")

solar\_radiation[[10]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2013")

solar\_radiation[[11]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2014")

solar\_radiation[[12]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2015")

solar\_radiation[[13]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2016")

solar\_radiation[[14]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2017")

solar\_radiation[[15]] <- upload(solar\_radiation\_folder, "Daily solar radiation", "2018")

solar\_radiation <- map(solar\_radiation, as\_tibble) %>% bind\_rows() %>% rename("solar\_radiation" = col\_name)

# max temp

max\_temp\_folder <- paste0(getwd(), "/Max temperatures - HKO/")

max\_temp <- list()

max\_temp[[1]] <- upload(max\_temp\_folder, "Max temp", "2004")

max\_temp[[2]] <- upload(max\_temp\_folder, "Max temp", "2004")

max\_temp[[3]] <- upload(max\_temp\_folder, "Max temp", "2006")

max\_temp[[4]] <- upload(max\_temp\_folder, "Max temp", "2007")

max\_temp[[5]] <- upload(max\_temp\_folder, "Max temp", "2008")

max\_temp[[6]] <- upload(max\_temp\_folder, "Max temp", "2009")

max\_temp[[7]] <- upload(max\_temp\_folder, "Max temp", "2010")

max\_temp[[8]] <- upload(max\_temp\_folder, "Max temp", "2011")

max\_temp[[9]] <- upload(max\_temp\_folder, "Max temp", "2012")

max\_temp[[10]] <- upload(max\_temp\_folder, "Max temp", "2013")

max\_temp[[11]] <- upload(max\_temp\_folder, "Max temp", "2014")

max\_temp[[12]] <- upload(max\_temp\_folder, "Max temp", "2015")

max\_temp[[13]] <- upload(max\_temp\_folder, "Max temp", "2016")

max\_temp[[14]] <- upload(max\_temp\_folder, "Max temp", "2017")

max\_temp[[15]] <- upload(max\_temp\_folder, "Max temp", "2018")

max\_temp <- map(max\_temp, as\_tibble) %>% bind\_rows() %>% rename("max\_temperature" = col\_name)

# mean temp

mean\_temp\_folder <- paste0(getwd(), "/Mean temperatures - HKO/")

mean\_temp <- list()

mean\_temp[[1]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2004")

mean\_temp[[2]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2005")

mean\_temp[[3]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2006")

mean\_temp[[4]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2007")

mean\_temp[[5]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2008")

mean\_temp[[6]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2009")

mean\_temp[[7]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2010")

mean\_temp[[8]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2011")

mean\_temp[[9]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2012")

mean\_temp[[10]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2013")

mean\_temp[[11]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2014")

mean\_temp[[12]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2015")

mean\_temp[[13]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2016")

mean\_temp[[14]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2017")

mean\_temp[[15]] <- upload(mean\_temp\_folder, "Daily mean temperature", "2018")

mean\_temp <- map(mean\_temp, as\_tibble) %>% bind\_rows() %>% rename("mean\_temperature" = col\_name)

# min temp

min\_temp\_folder <- paste0(getwd(), "/Min temperatures - HKO/")

min\_temp <- list()

min\_temp[[1]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2004")

min\_temp[[2]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2005")

min\_temp[[3]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2006")

min\_temp[[4]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2007")

min\_temp[[5]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2008")

min\_temp[[6]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2009")

min\_temp[[7]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2010")

min\_temp[[8]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2011")

min\_temp[[9]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2012")

min\_temp[[10]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2013")

min\_temp[[11]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2014")

min\_temp[[12]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2015")

min\_temp[[13]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2016")

min\_temp[[14]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2017")

min\_temp[[15]] <- upload(min\_temp\_folder, "Daily minimum temperature", "2018")

min\_temp <- map(min\_temp, as\_tibble) %>% bind\_rows() %>% rename("min\_temperature" = col\_name)

# total bright sunshine

total\_sunshine\_folder <- paste0(getwd(), "/Total bright sunshine (hours) - Kings Park/")

total\_sunshine <- list()

total\_sunshine[[1]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2004")

total\_sunshine[[2]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2005")

total\_sunshine[[3]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2006")

total\_sunshine[[4]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2007")

total\_sunshine[[5]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2008")

total\_sunshine[[6]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2009")

total\_sunshine[[7]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2010")

total\_sunshine[[8]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2011")

total\_sunshine[[9]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2012")

total\_sunshine[[10]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2013")

total\_sunshine[[11]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2014")

total\_sunshine[[12]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2015")

total\_sunshine[[13]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2016")

total\_sunshine[[14]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2017")

total\_sunshine[[15]] <- upload(total\_sunshine\_folder, "Total bright sunshine", "2018")

total\_sunshine <- map(total\_sunshine, as\_tibble) %>% bind\_rows() %>% rename("total\_bright\_sunshine" = col\_name)

# merging water quality and climatological data

d <- WaterQ %>% inner\_join(wind\_speed, by = "Dates")

d <- d %>% inner\_join(wind\_direction, by = "Dates")

d <- d %>% inner\_join(rainfall, by = "Dates")

d <- d %>% inner\_join(sea\_temp, by = "Dates")

d <- d %>% inner\_join(solar\_radiation, by = "Dates")

d <- d %>% inner\_join(max\_temp, by = "Dates")

d <- d %>% inner\_join(mean\_temp, by = "Dates")

d <- d %>% inner\_join(min\_temp, by = "Dates")

d <- d %>% inner\_join(total\_sunshine, by = "Dates")

# drop\_na

d <- d %>% drop\_na()

# creating a variable with the past day's date

d <- d %>% mutate(past\_day = Dates - ddays(1))

# past day E. coli average

past\_day\_ecoli\_average <- d %>%

group\_by(Dates) %>%

summarise(past\_day\_ecoli\_avg = mean(lnEcoli)) %>%

rename("past\_day" = Dates)

final <- d %>% inner\_join(past\_day\_ecoli\_average, by = "past\_day")

# past day solar radiation

past\_day\_solar\_rad\_avg <- d %>%

group\_by(Dates) %>%

summarise(past\_day\_sol\_rad\_avg = mean(solar\_radiation)) %>%

rename(past\_day = Dates)

final <- final %>% inner\_join(past\_day\_solar\_rad\_avg, by = "past\_day")

# past day total bright sunshine

past\_dat\_tot\_bright\_sunshine <- d %>%

group\_by(Dates) %>%

summarise(past\_tot\_bright\_sunshine = mean(total\_bright\_sunshine)) %>%

rename(past\_day = Dates)

final <- final %>% inner\_join(past\_day\_solar\_rad\_avg, by = "past\_day")

# Data analysis

analysis <- final %>% select(-c(Water\_zone, Station, Dates, Sample\_no, past\_day, Depth))

correlation <- cor(analysis, y = analysis$lnEcoli, method="pearson")

set.seed(1)

s <- analysis %>% initial\_time\_split(prop = 3/4)

training <- training(s)

testing <- testing(s)

crossV <- vfold\_cv(training)

# Best subsets

best\_subsets <- regsubsets(lnEcoli~., data = training, nvmax = 10, method = "exhaustive") %>%

tidy() %>%

filter(adj.r.squared == max(adj.r.squared))

# Recipe

f1 <- recipe(lnEcoli ~., data = training) %>% step\_normalize(all\_predictors()) #%>%

#step\_pca(all\_predictors(), num\_comp = tune())

# Recipe based on the best coeffiecients from best subsets analysis

f2 <- recipe(lnEcoli ~ lnBiochem\_ox\_demand + lnAmmonia\_nitrogen + lnDissolved\_ox + lnNitrate +

lnOrthophosphate\_phosphorus + lnSilica + sea\_temperature + max\_temperature +

past\_day\_ecoli\_avg,

data = training) %>%

step\_normalize(all\_predictors())

# Models

m1 <- linear\_reg() %>% set\_engine("lm")

m2 <- linear\_reg(penalty = tune(), mixture = 1) %>% set\_engine("glmnet")

m3 <- linear\_reg(penalty = tune(), mixture = 0) %>% set\_engine("glmnet") # Higher R squared with ridge

m4 <- decision\_tree(cost\_complexity = tune()) %>% set\_engine("rpart") %>% set\_mode("regression")

m5 <- bag\_tree(cost\_complexity = tune()) %>% set\_engine("rpart") %>% set\_mode("regression")

m6 <- boost\_tree(learn\_rate = tune()) %>% set\_engine("xgboost") %>% set\_mode("regression")

m7 <- rand\_forest(mtry = tune()) %>% set\_engine("ranger") %>% set\_mode("regression")

# Workflow

w <- workflow() %>% add\_recipe(f2) %>% add\_model(m1)

# Tuning

tuning <- w %>% tune\_grid(resamples = crossV, grid = 10)

tuning %>% collect\_metrics()

# Finalizing workflow, fitting the training set on predicting the testing set

w <- w %>% finalize\_workflow(tuning %>% select\_best("rsq"))

fit <- w %>% fit(training)

predict <- fit %>% predict(testing) %>%

bind\_cols(testing) %>%

filter\_all(all\_vars(!is.infinite(.pred))) %>%

rsq.n(.pred, lnEcoli, adj = TRUE)

predict