CAPSTONE PROJECT BY TEAM 08

CAPSTONE PROJECT BY TEAM 08 - A DATA SCIENCE APPROACH TO PREDICTING PHYTOPLANKTON LEVELS

Jinhao Huang (z5207964), Kurt Wang (z5207982), Ellen Wang (z5209313), Tal Weiner (z5209048), Connor (Pai) Yu (z5206136).

School of Mathematics and Statistics
UNSW Sydney

November 2020

Submitted in partial fulfilment of the requirements of the capstone course DATA3001

Plagiarism statement

I declare that this thesis is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere.

I acknowledge that the assessor of this thesis may, for the purpose of assessing it:

- Reproduce it and provide a copy to another member of the University; and/or,
- Communicate a copy of it to a plagiarism checking service (which may then retain a copy of it on its database for the purpose of future plagiarism checking).

I certify that I have read and understood the University Rules in respect of Student Academic Misconduct, and am aware of any potential plagiarism penalties which may apply.

By signing this declaration I am agreeing to the statements and conditions above.

Signed: Tal Weiner Date: 21/11/2020

Signed: Kurt Wang Date: 21/11/2020

Signed: Ellen Wang Date: 21/11/2020

Signed: Connor Gu Date: 21/11/2020

Signed: Jinhao zhuang Date: 21/11/2020

Acknowledgements

By far the greatest thanks must go to my supervisor for the guidance, care and support they provided.

Thanks must also go to Emily, Michelle, John and Alex who helped by proof-reading the document in the final stages of preparation.

Although I have not lived with them for a number of years, my family also deserve many thanks for their encouragement. Thanks go to Robert Taggart for allowing his thesis style to be shamelessly copied.

21/11/2020.

Abstract

This study assesses the Random Forest (RF) and General Linear Model (GLM) to forecast chlorophyll-a concentration off the coast of Forster, Australia given ocean properties as predictors. Performance of these models are assessed using Root Mean Square Error (RMSE) and AIC values. Chlorophyll-a concentrations have long been an indicator of phytoplankton abundance and biological growth in water. Phytoplankton abundance is directly correlated with biological growth as they form the basis of the food chain for all marine life. These numbers are of great interest to oceanographers, fisheries and biologists. Assessment of these models are necessary in order to continually regulate coastal waters and determine biological growth.

Contents

Chapter 1 Introduction 1

Chapter 2 Literature Review 2

Chapter 3 Material and Methods 6
- 3.1 Software . . . 6
- 3.2 Description of the Data . . . 6
- 3.3 Pre-processing Steps . . . 8
- 3.4 Data Cleaning . . . 8
- 3.5 Assumptions . . . 8
- 3.6 Modelling Methods . . . 9

Chapter 4 Exploratory Data Analysis 11

Chapter 5 Analysis and Results 14
- 5.1 A First Model - Random Forest . . . 14
- 5.2 A Second Model - Generalised Linear Model (GLM) . . . 17

Chapter 6 Discussion 21

Chapter 7 Conclusion and Further Issues 23

Appendix 26
- **Codes** . . . 26
  - 7.0.1 GLM Code . . . 26
  - 7.0.2 Ranger RandomForest Implementation . . . 34
- **Tables** . . . 35
  - 7.0.3 Reference Table for Variable abbreviations . . . 35

Chapter 1

Introduction

The measurement of chlorophyll-a in bodies of water has been a topic of interest in the research world for many years. Chlorophyll-a concentrations have long been an indicator of phytoplankton abundance and biological growth in water. These numbers are the base of the biological food chain in the ocean and are of great interest to oceanographers, fisheries, biologists amongst others. This study analyzes data collected by ocean gliders to determine the relationship between ocean properties and chlorophyll-a. The goal is to increase the amount of knowledge regarding predictive indicators of chlorophyll-a in Australian waters and how we could apply this to continually regulate the ocean whilst also determining biological growth.

Chapter 2

Literature Review

Chlorophyll-a concentrations are an essential part of the photosynthesis process where sunlight and carbon-dioxide (CO2) are transformed into energy (sugars) and oxygen (o2) [1]. It is important that these concentrations are monitored as they are directly correlated with phytoplankton abundance. Phytoplankton abundance is directly correlated with biological growth as they form the basis of the food chain for all marine life. This is demonstrated in "Relationship between Chlorophyll-a Concentration and Phytoplankton Biomass in Several Reservoirs in Czechoslovakia" and "Dynamic model of phytoplankton growth and acclimation: responses of the balanced growth rate and the chlorophyll a:carbon ratio to light, nutrient-limitation and temperature " where chlorophyll-a concentration results obtained from various reservoirs was in all cases positive The ability to predict these levels inform scientists of incoming anomalies in the ecosystem and how to i) prevent or ii) mitigate the impacts of such changes.Therefore, the ability to analyse chlorophyll-a concentrations from classic ocean properties form a crucial research question for researchers.

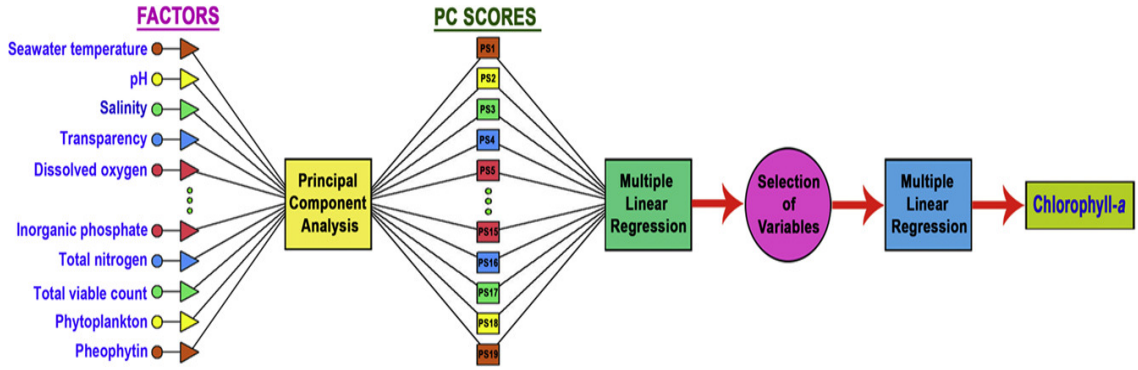
One manner in which chlorophyll-a levels is monitored is by observing pertinent and notable ocean properties such as nutrient levels present in water (Nitrate, Phosphate, Iron etc.), environmental conditions (salinity, wind), position of concentration (depth, lat, long, distance from surface). Ocean gliders are employed routinely to feed information back to researchers in order to model biological growth in terms of ocean properties.

As the result of such a crucial research question, there is a plethora of resources that have attempted to best model and predict chlorophyll-a concentrations in marine/coastal environments. The two primary models are GLM and black-box methods, a summary of their approaches and methodologies have been listed below:

The general linear model is a useful and simple framework for understanding how independent variables change with the dependent variable that have distributions that are not necessarily normal. It's an intuitive method that can help to explain ocean properties and chlorophyll-a concentrations. This methodology has been applied in multiple scenarios, the first of which is a simple GLM with multiple predictors."A Regression Model for the Prediction of Chlorophyll a in Lake Okeechobee, Florida 2009" shows a brief comparison between a simple linear model vs. the general linear model shows us that taking the logarithm of our dependent variable chlorophyll-a and logarithm of significant variables date, total phosphorus, interaction of differing zones in the lake, nitrogen and temperature gives us a R^2 value of 27.6%. This models' strength lies in the fact that individual independent values also underwent transformations to best fit the model. Interaction terms were also analysed and included under the assumption of a linear relationship. Whilst

this model intuitively explains the relationship between chlorophyll-a and other predictors, there was significant uncertainty of interpretations of some of the parameters and it might be worthwhile to refine these models to produce a higher R-squared value [2].

Another application of the GLM is preceded by PCA analysis to choose relevant predictors before feeding it into the model in "A novel approach to predict chlorophyll-a in coastal-marine ecosystems using multiple linear regression and principal component scores". The resulting predictors of seawater temperature, pH, salinity, dissolved oxygen, nitrate produced a model with a predictive success rate of 83.8%. https://www.sciencedirect.com/science/article/abs/pii/S0025326X20300205

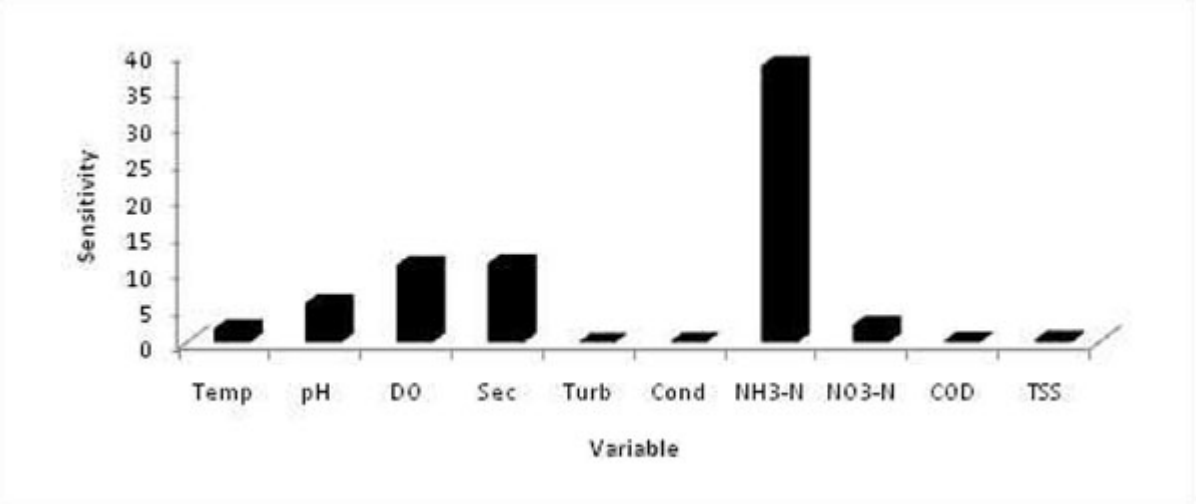


Many attempts have been made to model chlorophyll-a concentrations with state-of-the-art technology. Some of the more popular methods include: Discrete Wavelet Transformation/Artificial Neural Network (ANN), Random Forest, Hybrid Evolutionary Algorithm and Fuzzy Logic.

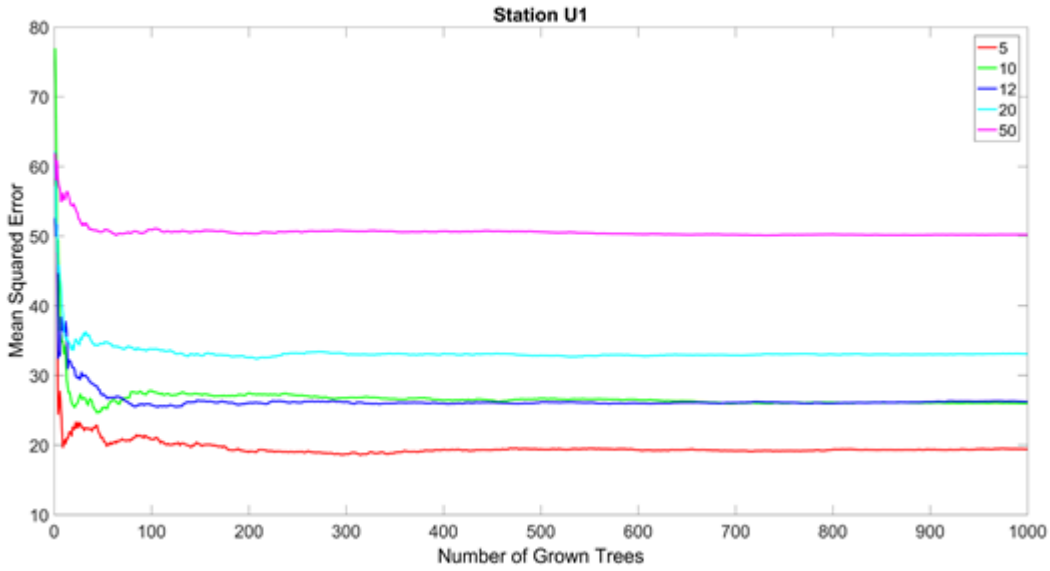
Artificial neural network (ANN) models combined with Discrete Wavelet Transformations are highly flexible function approximators that can be used to model nonlinear relationships between independent and dependent variables. In "ANN_Ensemble_Method_ an ensemble of ANN models was created to perform autocorrelation analysis between time-series data in order to produce short-term forecasts. The R-squared and RMSE values were used to assess models and produced an overall improvement to a best single model with Discrete Wavelet Transformation input. In a similar paper, "ANN_MLR_WDT_Rajaee_Boroumand2015", the data was transformed by the Discrete Wavelet Transformation and fed into an Artificial Neural Network. On top of RMSE, the Nash-Sutcliffe model efficiency coefficient (E) was also used as a statistic to measure accuracy of the model. Up to 15 layers were used in the Backpropagation network of the Artificial Neural Network to predict chlorophyll-a levels. ANN models have proven to be successful when applied to predict chlorophyll-a concentration in coastal water bodies. https://www.tandfonline.com/doi/full/10.1080/19942060.2018.1553742 https://www-sciencedirect-com.wwwproxy1.library.unsw.edu.au/science/article/pii/S0141118715001157

More context specific models such as Fuzzy Logic and Hybrid Evolutionary Algorithms are popular modelling methodologies used by biologists to model relationships in coastal ecosystems and genomics. This is shown in "Assessment of predictive models for chlorophyll-a concentration of a tropical lake". A sensitivity analysis was done on each of the input variables against chlorophyll-a before

using RMSE to measure level of prediction error, correlation coefficients and Area Under Curve (AUC) criteria to assess the models. The selected variables using HEA were pH, depth, dissolved oxygen and nitrate nitrogen. The selected variables using FL were temperature, pH, dissolved oxygen, ammonia nitrogen, nitrate nitrogen and depth. HEA performed the best in terms of assessment criteria. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3278828/



The last black-box method is the Random Forest. Random Forests are a foundational ensemble learning method used to make decisions for feature engineering. There is no need for a prior determination of initial assumptions and have the ability to select a small number of significant parameters given many. The RF model allows us to forecast trends in time series datasets. "Application of the Random Forest model for chlorophyll-a forecasts in fresh and brackish water bodies in Japan, using multivariate long-term databases 2017" uses RMSE as assessment criteria to determine the parameters required to build a forecasting model.



https://www.sciencedirect.com/topics/engineering/random-forest https://iwaponline.com/jh/article/20/1/206/37900/Application-of-the-Random-Forest-mo

The key downsides to using black-box methods are that results are difficult to interpret as the methods are not developed alongside the context of the problem. These methods are constantly being updated and improved and fall out of use frequently. There is also significant conflicting evidence when it comes to using black-box methods as they have a tendency to overfit models as opposed to providing a good understanding of the dataset at hand. Overall, significant research should be done on both the method, context and understanding of measuring statistics (such as RMSE, R-squared, ROC, AUC, correlation coefficients) before modelling.

The general conclusions being drawn by these papers show that the relationship between chlorophyll-a concentrations and ocean properties can be modelled and are indicative of predicting and modelling biological growth. The primary variables that are being included in the model are: date/time, pH, depth, dissolved oxygen and nitrogen. Alternative viewpoints and variables being included in papers vary based on location and methodology and should be factored into modelling approach.

Chapter 3

Material and Methods

3.1 Software

R is the primary software for this project. The decision was made as the code (anfog.R) which contained general data analysis and add-on files (ncParse.R) provided to load the dataset was written in R. ANFOG is a facility of Australia's Integrated Marine Observing System and is responsible for the operation and maintenance of the ocean glider fleet.

An E480 Lenovo Thinkpad with a Windows operating system (64-bit) and intel Core i7 8th Gen as the Processor was used to run the models. The random forest model took approximately 33 minutes to run locally. It's likely that the time it took to train this model would have been vastly improved upon given better equipment and more RAM.

The libraries used in R version 4.0.0 are:

- Tidyverse (1.3.0)
- Magrittr (1.5)
- Readxl (1.3.1)
- Mgcv (1.8.31)
- MASS (7.3.51.5)
- Caret (6.0.86)
- RandomForest (4.6.14)
- Ranger (0.12.1)

3.2 Description of the Data

The data for this problem is provided by ocean gliders who collect 10,000s of measurements of different characteristics of the water, such as the pressure, temperature, salinity, dissolved oxygen and the concentration of chlorophyll-a (fluorescence). Data is collected per mission, which last around 21 days on average. Data measurements are collected every 2 seconds, then they are transmitted from the gliders and passed through for processing and quality control procedures. Each entry into the files is ordered by its timestamp and stored in a timeseries. Each entry includes data for around 30 variables with millions of observations being made per variable in total.

These ocean gliders are deployed in cities off the east coast of Australia including but not limited to Bass Strait, BCG, Cairns, Charlotte Bay, Forster, Harrington. The data used for this project will be limited to the eleven missions completed in the Forster area (including the most recent release in October).

OPeNDAP (Open-source Project for a Network Data Access Protocol) is the developer that enables researchers and scientists to share data more easily over the internet.

Data Access portal used for this study: http://thredds.aodn.org.au/thredds/dodsC/IMOS/ANFOG/slocum_glider/Forster20170911/IMOS_ANFOG_BCEOPSTUV_20170911T0
SL287_FV01_timeseries_END-20171002T010328Z.nc.html

Given the large size of the data, there was an option to select relevant variables through the data access portal on OPeNDAP. The URL link with query is fed into R code, which downloads data for use.

The sizes of the data files vary around the 500kb mark. Each file contains about 400,000 data points and is representative of the number of data points collected per mission.

A look at summary(dataset) shows that our data is stored in three distinct groups

```
# > summary(dataset)
#            Length Class  Mode
# metadata   48     -none- list
# dimensions  2     -none- list
# variables  30     -none- list
```

The variables subsection is the factor that interests us the most. Displaying summary(dataset$variables), comments were added for interpretability:

```
# > summary(dataset$variables)
#            Length Class  Mode Comments
# PLATFORM   13     -none- list N/A
# DEPLOYMENT 14     -none- list N/A
# SENSOR1     9     -none- list Type CTD: Measures conductivity, temperature,
# SENSOR2     9     -none- list Type Eco Puck: Backscattering & fluorescence
# SENSOR3     9     -none- list Type Oxygen Sensor
# SENSOR4     9     -none- list Type Multispectral radiometer: measures light
# LATITUDE   27     -none- list Y axis
# LONGITUDE  27     -none- list X axis
# HEAD       25     -none- list Vehicle degrees: clockwise from magnetic nort
# UCUR       27     -none- list Eastward velocity of seawater: between surfac
# VCUR       27     -none- list Northward velocity of seawater: between surfa
# UCUR_GPS   25     -none- list Eastward velocity of seawater: drift between
# VCUR_GPS   25     -none- list Northward velocity of seawater: drift betweer
# PHASE      26     -none- list Phase of trajectory*
# PROFILE    25     -none- list Phases are given a profile number at each cha
# PRES       27     -none- list Pressure of seawater: measured by CTD
# DEPTH      28     -none- list Depth of seawater by metre
# TEMP       26     -none- list Temperature of seawater by celsius
# CNDC       26     -none- list Electrical conductivity
# PSAL       26     -none- list Salinity of water using Gibbs-SeaWater
# DOX2       26     -none- list Moles of oxygen per unit mass
# DOX1       26     -none- list Mole concentration of dissolved molecular oxy
```

```
# CPHL        26      -none- list TARGET: mass concentration of chlorophyll in
# CDOM        24      -none- list Concentration of coloured dissolved organic m
# VBSC        24      -none- list Volume scattering function
# BBP         24      -none- list Particle backscattering coefficient
# IRRAD443    24      -none- list Downwelling spectral irradiance in seawater (
# IRRAD490    24      -none- list Downwelling spectral irradiance in seawater (
# IRRAD555    24      -none- list Downwelling spectral irradiance in seawater (
# IRRAD670    24      -none- list Downwelling spectral irradiance in seawater (
```

Code	Meaning	Comment
0	Surface drift	Glider is drifting on the surface layer
1	Descent	Glider is descending
2	Subsurface drift	Glider is drifting in subsurface
3	Inflexion	Glider is changing its trajectory
4	Ascent	Glider is ascending
5	Grounded	Glider touched the ground or seafloor or onshore
6	Inconsistent	Glider pressure is not consistent with the surrounding pressures

*Reference table for phases of the glider trajectory A glider regularly performs surface, descent, inflexion, subsurface drift and ascent phases. During ascent or descent phase, the glider performs vertical profiles:

3.3 Pre-processing Steps

The observations from dataset variables from OPeNDAP Forster missions 2017-2018 were collated into a csv file and 2019 into a separate csv file. A data frame was created so that all variables are accessible and easy to use. The variable names have been changed for the sake of readability. The data was then split into train-test (66:33 - 2017/2018 on 2019 data) csv's for model training and testing.

3.4 Data Cleaning

Initially, the whole dataset was observed by using the summary and head functions. However, it was noticeable that some data points were significantly more reliable than others due to the number of NULL values present. Only data points where quality control check flags were equal to 1 were retained. This meant that the data was of good quality.The remaining NULL values were removed and this cleaned dataset was used for training/testing.

3.5 Assumptions

The main assumption we are making on the data is that each measurement collected by the ocean glider sensors is independent of all other measurements. This ensures our response variables are independent.

3.6 Modelling Methods

The first model needed to be robust, quick to get results and didn't need too much tuning, as well as having good predictive ability. Being the first model, the Data Science Iterative workflow process will be adhered to so after considerations of this initial model, analysing results as well as strengths and weaknesses it can then be determined if other models are needed to improve or validate results.

The first model that was chosen was a Random Forest implementation based in R. This choice was based on a range of scientific research articles where overwhelming research pointed towards Random Forest regressors [3]. Due to the nature of how Random Forest regressors work and the inherent 'randomness' incorporated into the model it is very effective when it comes to predictive modelling out performing most regression models while also being robust to outliers, working well with non-linear data, lower risk of overfitting and efficiency on larger datasets when predicting [4]. Moreover Random Forest models are remarkably good "out-of-the-box" and require little tuning which serves as a baseline level of prediction accuracy for future models. In addition to this existing implementations (functions) in R such as "randomForest()" and "ranger()" include built-in validation sets so there is no need to sacrifice additional data for testing [5]. There are also negatives to the Random Forest implementation which is ultimately why it was not used as the final model, these will be discussed in conjunction with the examples and observations from implementation below.

The second model we have chosen is a Generalized Linear Model (GLM). The GLM is made up of 3 components:

1. The Random Component - response variables $Y_1, \ldots, Y_n$ are independent of one another and belong to a distribution from the exponential family.
2. The Systematic Component - there exists a linear predictor $\eta = X\beta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$.
3. The Link Function - a function $g()$ that specifies how the expected value of the response ($\mu = E[Y_i]$) relates to the linear predictor (η), given by $g(\mu) = \eta$.

The GLM does not require the response to be normally distributed nor for there to be a linear relationship between the response and each independent variable x_i. However the GLM does require a linear relationship between the transformed mean of the response and the linear predictor via the link function.

To determine which distribution best models the response variable, a histogram of the response variable chlorophyll-a concentration was plotted. As shown below, the data is heavily right skewed, ranging between (0, 8) with majority of observations closer to 0. Using method of moments estimation for finding the parameters of a Gamma Distribution, a gamma density curve is plotted over the histogram of chlorophyll-a. As shown below, the plot suggests that a Gamma distribution could fit the data.

Choosing the Gamma Distribution, the model is of the form:

$$cphl \sim Gamma(k, \theta)$$

$$E[cphl|X] = k\theta = \mu$$

$$\mu = g^{-1}(\eta)$$

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_p x_p p$$

where we have p predictor variables and $g^{-1}()$ is the inverse of our chosen link function. The initial link function chosen was the canonical link function, which for the gamma distribution is the inverse function. Hence,

$$g(\mu) = \frac{1}{\mu}\frac{1}{\mu} = \eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_p x_p.$$

The log link function was also considered. In this case, $g(\mu) = log(\mu)$.

The generalised linear model was subject to a variable selection function, StepAIC. A subset of explanatory variables need to be chosen to produce the model with the best fit; the lowest Information Criterion and root mean squared error (RMSE) values. StepAIC is a stepwise regression function that utilises the Akaike Information Criteria and quantifies model complexity and goodness of fit, through the formula of $AIC = 2 * parameters - 2 * likelihoodfunction$. As suggested by the formula, the less information lost by variable, the lower the AIC value and the higher quality the subset of variables chosen. AIC also minimises overfitting and underfitting by selecting the subset of variables this way. Due to distribution of variables that had a lot of values close to 0, the stepAIC function in R became faulty and a brute force forward selection AIC method was implemented instead through comparing models and their AIC value at each step. An excerpt of the code of the method and the selection of variables is below.

```
fit3 <- glm(cphl~depth, start = rep(1,2), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.1 <- glm(cphl~depth+temp, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.2 <- glm(cphl~depth+psal, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.3 <- glm(cphl~depth+dox1, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.4 <- glm(cphl~depth+cdom, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.5 <- glm(cphl~depth+vbsc, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.6 <- glm(cphl~depth+irrad443, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.7 <- glm(cphl~depth+irrad490, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.8 <- glm(cphl~depth+irrad555, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)
fit3.9 <- glm(cphl~depth+irrad670, start = rep(1,3), family = Gamma(link = "inverse"), data = model_data_avg5_pos)

aic3.1 <- AIC(fit3.1)
aic3.2 <- AIC(fit3.2)
aic3.3 <- AIC(fit3.3)
aic3.4 <- AIC(fit3.4)
aic3.5 <- AIC(fit3.5)
aic3.6 <- AIC(fit3.6) ### WINNER
aic3.7 <- AIC(fit3.7)
aic3.8 <- AIC(fit3.8)
aic3.9 <- AIC(fit3.9)

min3 <- min(c(aic3.1,aic3.2,aic3.3,aic3.4,aic3.5,aic3.6,aic3.7,aic3.8,aic3.9))
(min3 == c(aic3.1,aic3.2,aic3.3,aic3.4,aic3.5,aic3.6,aic3.7,aic3.8,aic3.9))
```

```
> min3 <- min(c(aic3.1,aic3.2,aic3.3,aic3.4,aic3.5,aic3.6,aic3.7,aic3.8,aic3.9))
> (min3 == c(aic3.1,aic3.2,aic3.3,aic3.4,aic3.5,aic3.6,aic3.7,aic3.8,aic3.9))
[1] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
```

Chapter 4

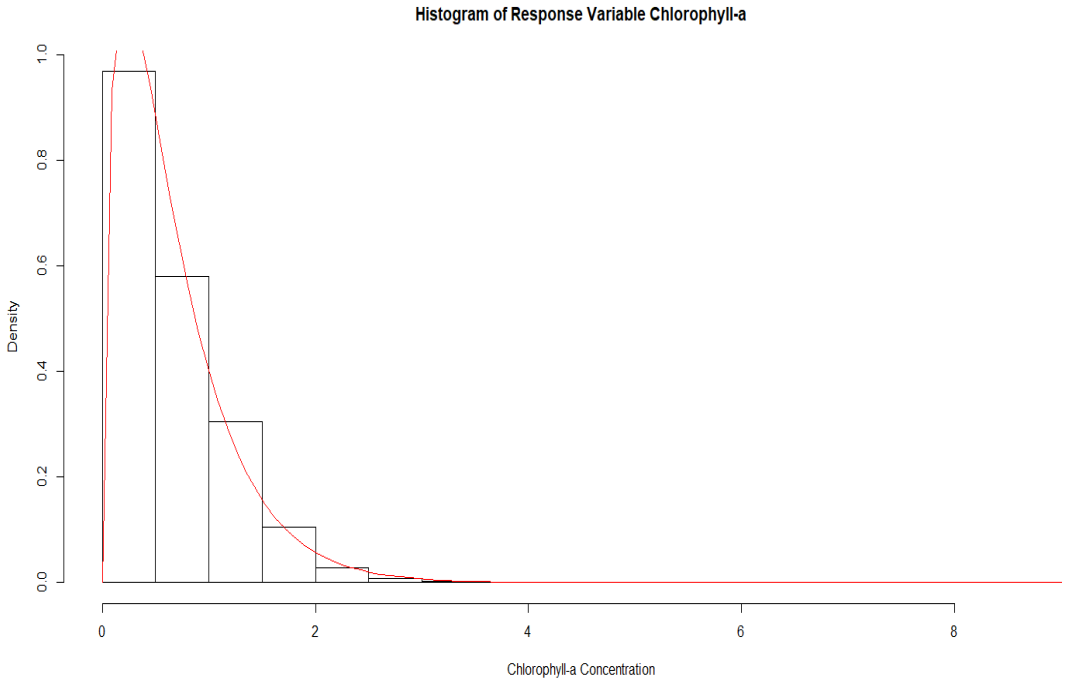
Exploratory Data Analysis

The initial variables in the dataset consisted of platform, deployment and sensors which are qualitative variables that do not help explain chlorophyll growth so are excluded from being used. Furthermore, any variables which are related to glider usage, such as phase, are excluded as they do not contribute towards our research question of using 'ocean properties'. Two further variables, latitude and longitude are also excluded because the research question does not include location as one of the properties in determining chlorophyll growth. The variable 'dox2' had many missing data values and 'dox1' explained the same variable of dissolved oxygen so 'dox2' was excluded. Finally, velocity of seawater was excluded as a time series model was not used and velocity did not add value to the model as an ocean property.

The remaining variables consists of chlorophyll-a, depth, temperature, sea water electrical conductivity, salinity, dissolved oxygen, volume scattering function, concentration of dissolved organic matter and four different wavelengths of spectral irradiances in water. As the variable of chlorophyll was our measure of predicting biological growth, this was regressed against all the other variables initially using a general linear model in Rstudio.

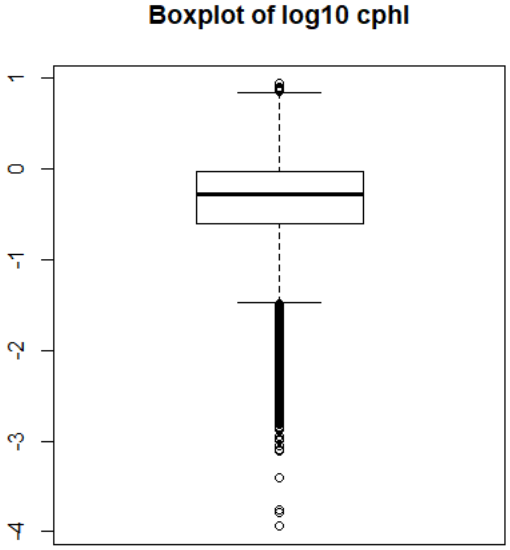
From the Forster missions, a selection of four different data sets were made, each from different seasons, to ensure there was no bias with different seasons affecting biological matter in the water. Due to a large amount of data points, the average of five values for all variables were taken. This also prevented any null values which would have caused problems in regressing in a generalised linear model as certain family distributions like gamma only accept positive values [6].

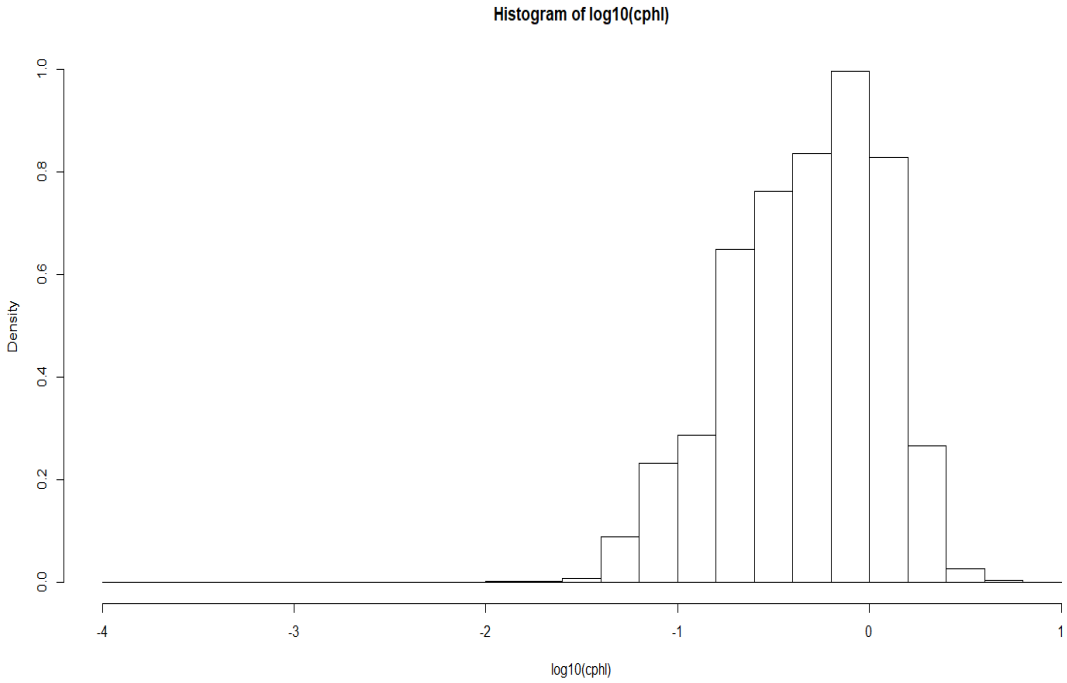
Taking a histogram of chlorophyll-a values in our training set, a distribution similar to that of gamma is observed as shown in below.



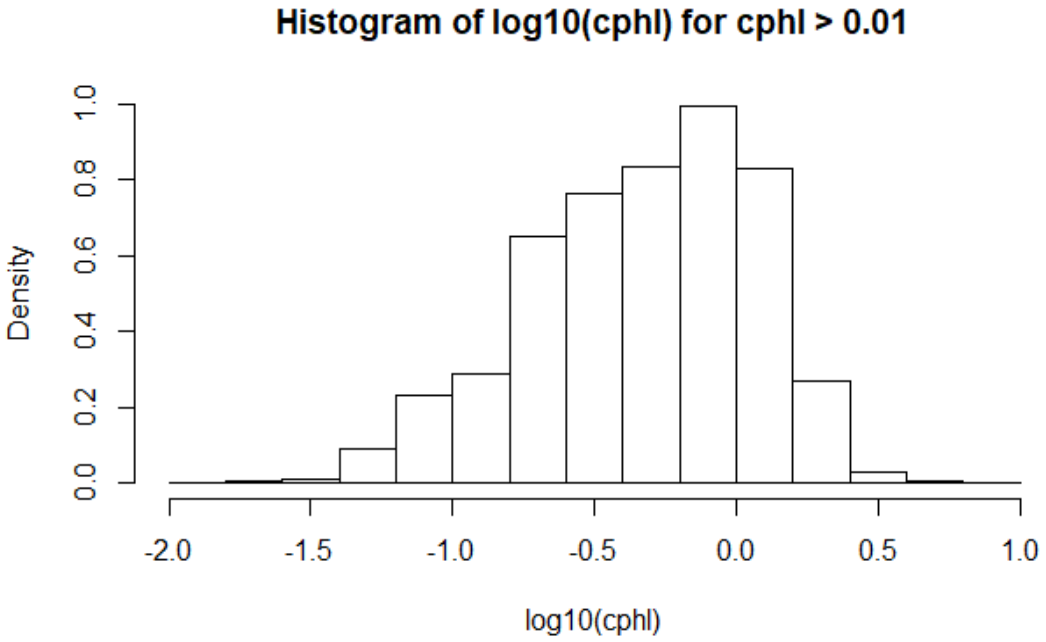
For the gamma family distribution, method of moments of estimation was applied to gain shape and scale parameters for the overlay of the gamma distribution. This gave an accurate curvature distribution shown by the red line, going through the middle of each histogram bin which highlights why the gamma family distribution was considered for the GLM.

A boxplot and histogram of log10(cphl) was utilised to demonstrate the significantly low values for chlorophyll collected. Chlorophyll values below 0.01 were removed due to significant issues arising in the use of a gamma distribution with values so close to zero.





After excluding those outliers, the distribution of log10(chlorophyll) is noted below with less skew, suggesting remaining values will work better for the gamma GLM.



Chapter 5

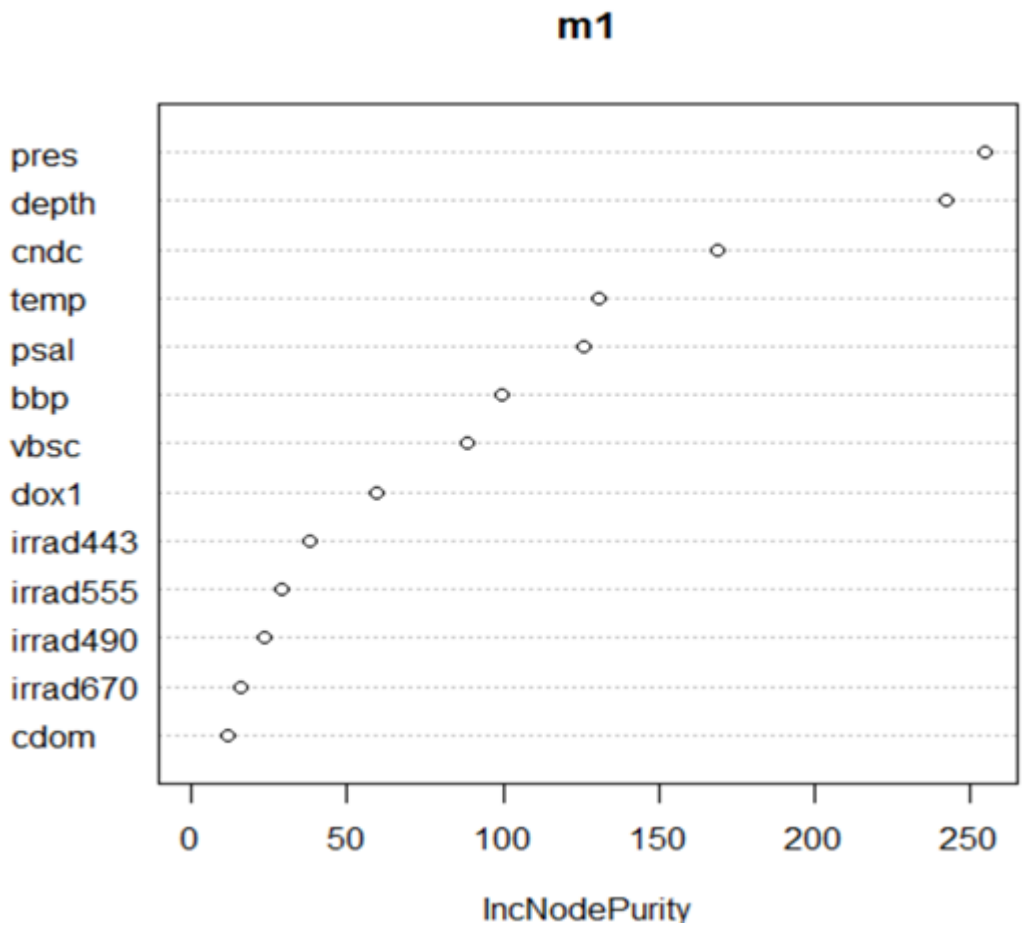
Analysis and Results

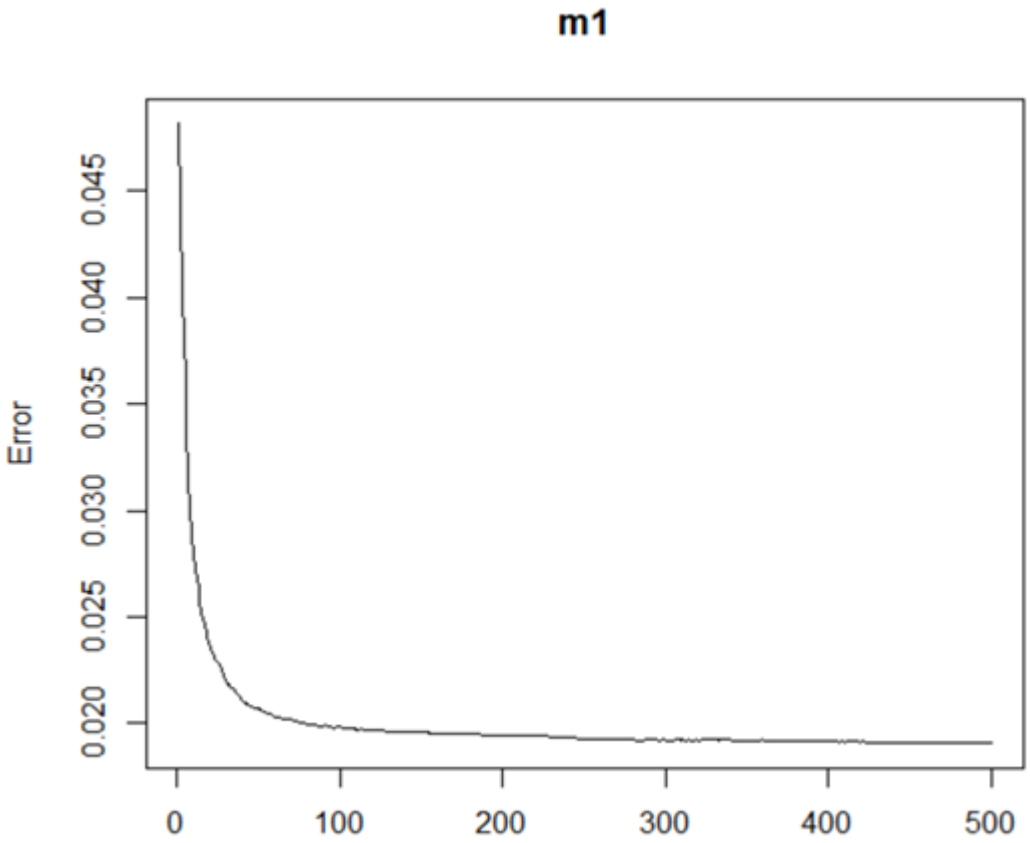
5.1 A First Model - Random Forest

Utilising "randomForest()" as a baseline predictive model with basic implementation as follows:

```
# > m1_rf_fit <- randomForest(formula = cphl~., data = m1)
# > varImpPlot(m1_rf_fit)
# > plot(m1_rf_fit)
```

Where 'm1' is the model, 'cphl' is the target variable and 'm1' is the dataset being used. (Note here, as mentioned previously in the data handling section, there were multiple data sets varying over a range of times, this example is just a representation of the whole). In terms of training the model on each data set (containing ~400,000 rows and 14 variables each) the resource requirements were enormous and unsustainable with hours of training time and over 4GB of memory allocation required. This was unsurprising as referenced in "Journal of Statistical Software" [7] where runtime for training would exceed 100 hours and use up to 39.05 GB of memory. For a similar data set with mtry =3 this was scalable by comparison. In order to be able to train the model, smaller randomised datasets were extracted from the mother-set to provide a k-folds validation-esque implementation. Experimentation proved that a value of around 8000 observations with the same 14 variables worked well and was within the limitations of resourcing constraints.


m1
pres
depth
cndc
temp
psal
bbp
vbsc
dox1
irrad443
irrad555
irrad490
irrad670
cdom
0
50
100
150
200
250
IncNodePurity



From Figure 1, this is a plot of variable importance based on the Gini impurity index used for calculating the splits in trees. The higher the value of mean decreases accuracy or mean decrease in Gini score, the higher the importance of the variable to our model [8]. It is clear to see the variables 'pres', 'depth' are by far the most important variables with 'cndc' , 'temp', 'psal', 'bbp' and 'vbsc' the others with moderate importance. This comes to no surprise in terms of feature selection as previous research concluded these variables (Pressure, depth, temperature etc.) have a high effect and positive correlation with phytoplankton growth and thus chlorophyll levels [9]. Furthermore in Figure 2 a plot of errors vs. ntrees (number of trees) it is clear that the errors drop sharply and stabilise around the 100 tree mark, but further analysis yields that 477 trees produces the minimum error. In order to first mitigate the number of variables used in the training problem, a faster implementation of Random Forest regression was used – "ranger()" which is over 100 times faster and ˜160 times more memory efficient [7].

```
#> m1_fit <- ranger(cphl ~ ., data = m1)
```

Type:	Regression
Number of Trees:	500
Sample Size:	389675
Number of independent variables:	13
Mtry:	3
Target node size:	5
Variable importance mode:	none

Type:	Regression
Splitrule:	variance
OOB prediction error (MSE):	0.003912073
R square (OOB):	0.9842886

From the output of the more efficient ranger() implementation, the sample size of observations was drastically increased to just under 400000 variables. This meant that all observations for each dataset was able to be taken into account in the model. Again from the output above (an instance of all the datasets tried) the OOB prediction error (MSE) is around 0.003912073 which signifies a good predictor. Other datasets produced very similar results with a maximum deviation of ˜0.0015.

Although the predictive capabilities are very good in terms of accuracy it doesn't help as much to answer the research question. Where the randomness of the algorithm shines through it is also one of the downfalls of this type of model. Not only is it slow in training on large datasets; conditioned on resources such as time, computing power, electricity, costs but the major disadvantages here are:

1 Black Box Approach

2 Extrapolation Problem

Black Box Approach: Random Forest implementations are essentially a black box approach for modern statisticians as there is little control over what the model does. Although different parameters, seeds and training data can vary the models performance to a degree, it is not significant in most cases. Furthermore as the implementation process involves a large number of trees (most of the time deep) and each tree is trained on bagged random selection of features it is not feasible to examine each tree in order to gain an understanding of the whole process [10]. This makes it almost impossible to fully grasp the inner workings of the model and not as applicable to the research question – in determining which common ocean properties can be used to predict chlorophyll-a levels.

Extrapolation Problem: Another serious problem in the scope of the research questions is the ability of the model to extrapolate and make predictions outside of the train/test data set. Essentially when the regressors "...are tasked with the problem of predicting values not previously seen, it will always predict an average of the values seen previously. Obviously the average of a sample cannot fall outside the highest and lowest values in the sample." [11]. This makes the model unfeasible for the target question as variables such as pressure, temperature, salinity will almost be certain in future observations that fall outside of currently 'seen' ranges and hence an incorrect prediction will be made.

Due to the shortcomings of this model albeit the great predictive ability further models were experimented with varying degrees of success and results.

5.2 A Second Model - Generalised Linear Model (GLM)

After finding the correct family distribution and canonical link function, the generalised linear model was subject to a variable selection function, StepAIC. A subset of explanatory variables need to be chosen to produce the model with the best fit; the lowest Information Criterion and RMSE values. StepAIC is a stepwise regression function that utilises the Akaike Information Criteria and

quantifies model complexity and goodness of fit, through the formula of $AIC = 2 \times parameters - 2 \times likelihoodfunction$.

Starting with the null model with just the intercept, variables were added onto the model until the AIC value was at its lowest. This concluded with the only variable 'irrad-670' being excluded from the model.

A final three models were chosen. The full model, the model without 'irrad670' and the model with link 'log' were tested against four different seasons of four different datasets to calculate the models' respective root mean squared error.

```
> gamma_invfull.rmse
[1] 0.5275570 0.5653603 3.4657598 0.3054035
> gamma_invno670.rmse
[1]  0.5399932  0.5635647 31.5648761  0.3053689
> gamma_log.rmse
[1] 3.955525e+10 4.412198e-01 9.371040e+05 2.909685e-01
```

The model without irrad670 and inverse link function had the lowest AIC and second lowest RMSE. The full model GLM consisting of all the variables had the lowest RMSE, however the reduction in deviance between the model without irrad670 and the full model was not deemed significant enough to include irrad670. Furthermore, irrad670 contained an insignificant p-value in the full model, adding more reason to go with the model without it.

```
Call:
glm(formula = cphl ~ ., family = Gamma(link = "inverse"), data = model_data_avg5_pos,
    start = rep(1, 12))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6568  -0.5156  -0.1387   0.2761   3.4298

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)  1.807e+02  7.215e-01  250.456   <2e-16 ***
depth        2.070e-02  8.545e-05  242.191   <2e-16 ***
temp        -9.170e+00  3.307e-02 -277.280   <2e-16 ***
cndc         9.002e+01  3.238e-01  278.019   <2e-16 ***
psal        -1.217e+01  4.497e-02 -270.602   <2e-16 ***
dox1        -4.750e-03  7.785e-05  -61.009   <2e-16 ***
cdom        -4.100e-01  3.672e-03 -111.652   <2e-16 ***
vbsc        -1.403e+02  9.143e+00  -15.343   <2e-16 ***
irrad443     7.016e-02  6.486e-04  108.179   <2e-16 ***
irrad490    -5.644e-02  9.272e-04  -60.875   <2e-16 ***
irrad555     3.037e-02  9.105e-04   33.355   <2e-16 ***
irrad670     1.235e-03  1.878e-03    0.658    0.511
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.3608169)

    Null deviance: 216776  on 308590  degrees of freedom
Residual deviance: 108041  on 308579  degrees of freedom
AIC: 97926

Number of Fisher Scoring iterations: 23
```

The final model is of the following form:

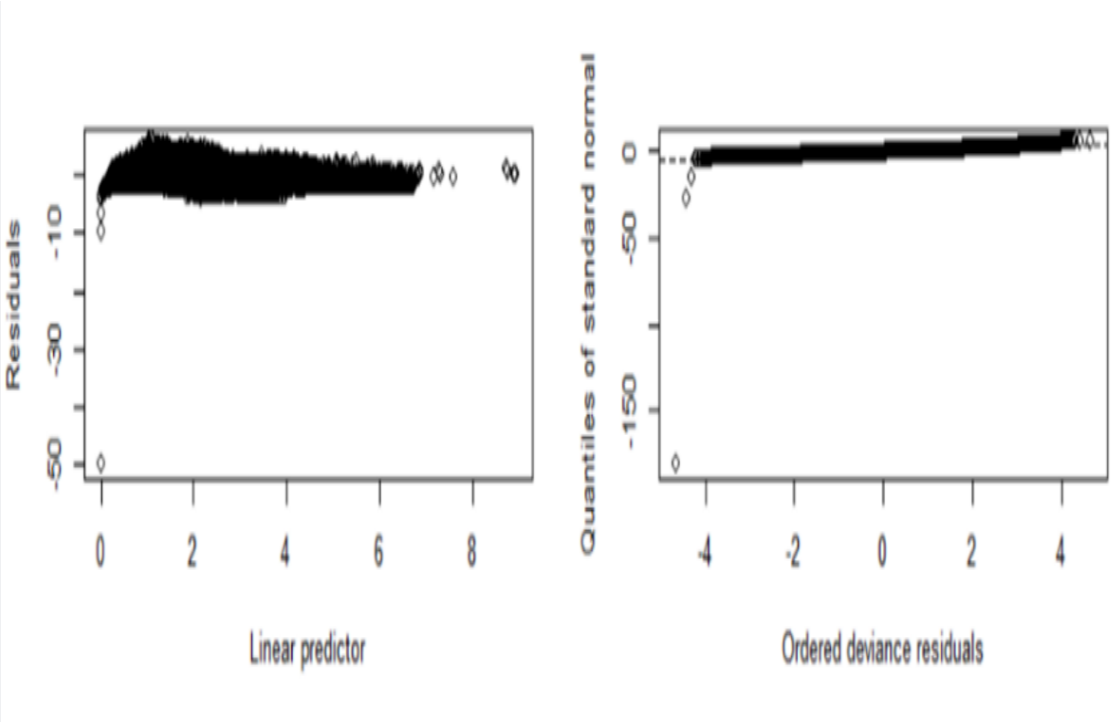
$$\frac{1}{\mu} = \beta_0 + \beta_1 depth + \beta_2 temp + \beta_3 cndc + \beta_4 psal + \beta_5 dox1 + \beta_6 cdom + \beta_7 vbsc + \beta_8 irrad443 + \beta_9 irrad490$$

with the β coefficient estimates shown below:

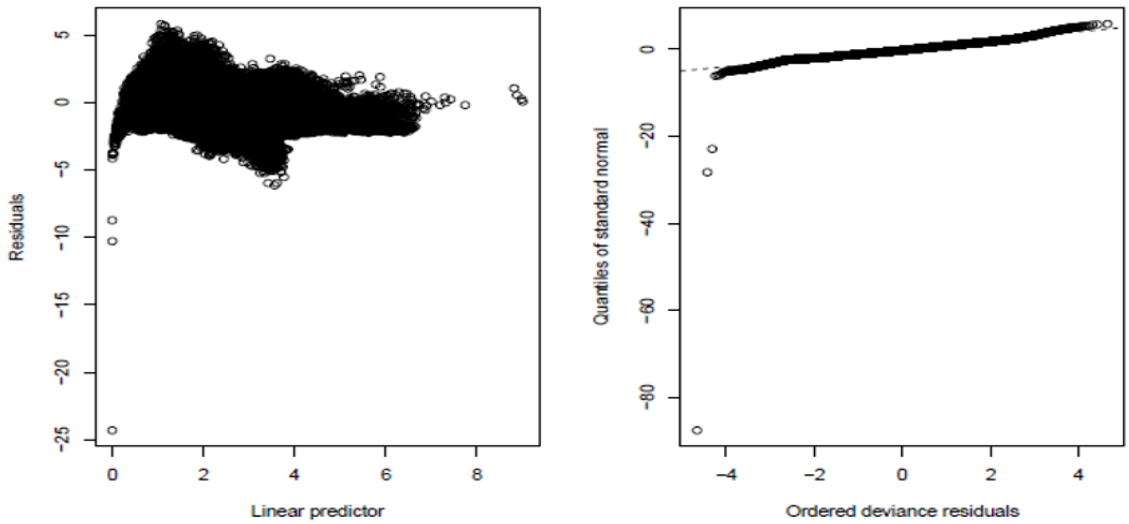
```
Call:
glm(formula = cphl ~ . - irrad670, family = Gamma(link = "inverse"),
    data = model_data_avg5_pos, start = rep(1, 11))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6570  -0.5156  -0.1388   0.2762   3.4296

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.807e+02  7.214e-01  250.50   <2e-16 ***
depth        2.070e-02  8.536e-05  242.47   <2e-16 ***
temp        -9.170e+00  3.306e-02 -277.36   <2e-16 ***
cndc         9.001e+01  3.237e-01  278.09   <2e-16 ***
psal        -1.217e+01  4.496e-02 -270.66   <2e-16 ***
dox1        -4.753e-03  7.774e-05  -61.13   <2e-16 ***
cdom        -4.100e-01  3.671e-03 -111.69   <2e-16 ***
vbsc        -1.405e+02  9.138e+00  -15.37   <2e-16 ***
irrad443     7.029e-02  6.187e-04  113.62   <2e-16 ***
irrad490    -5.671e-02  8.324e-04  -68.13   <2e-16 ***
irrad555     3.073e-02  7.239e-04   42.46   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.3608488)

    Null deviance: 216776  on 308590  degrees of freedom
Residual deviance: 108041  on 308580  degrees of freedom
AIC: 97925

Number of Fisher Scoring iterations: 24
```

Using the boot library in R, we utilised the glm.diag and glm.diag.plots functions to assess the final model's fit. The plot uses the deviance residuals of the GLM. [12]

The left graph plots the deviance residuals against the fitted values and the right graph plots a normal QQ plot of the standardized deviance residuals. Both graphs demonstrate a large number of predictor data points with extreme residuals.



When looking into these points to determine why they had such large residuals, none of the predictor values for these observations suggested a reason for the large residual. Even after removing the obvious outliers, more appeared.



These high residuals highlight problems in predicting values that occurred with the final generalised linear model.

Chapter 6

Discussion

Both the Random Forest Model (RFM) and the final Generalised Linear Model (GLM) trained from the data show positive results in highlighting the significance of certain oceanic properties in determining Chlorophyll-a levels. Although the Random Forest Model has an excellent "out-of-the-box" performance with little tuning and preprocessing required, the GLM still performed better overall taking into account efficiency and interpretability. Due to the dataset typically containing several outliers / extreme values, the main advantage of the RFM lies in its lack of extensive data cleaning required, as the built-in validation and robust outlier detection ensures accurate prediction. However, the model requires significant memory usage (4GB for each dataset) and becomes increasingly slow with scalability, which may defer its usefulness to marine biologists and scientists. Furthermore, the model is unable to provide detailed interpretability and information on each individual predictor. Thus, the GLM with Gamma distribution was chosen. This choice is also supported by [12] which used a linear model to predict chlorophyll-a levels in Lake Okeechobee using a few similar predictors such as temperature. However, the Random Forest is still a good baseline model for comparison.

For the Generalized Linear Model with Gamma distribution, two potential link functions were considered: inverse (canonical) and log. Through a variety of deviance and residuals analysis as well as a manual stepwise model selection process using AIC, the inverse link function was decided to be the most effective. Firstly, in the model with an inverse link function "irrad670" displayed an unaccepted P-value, suggesting insignificance in determining chlorophyll-a levels. Upon analysis of deviance test between the GLMs including and without "irrad670", it's removal was justified as the deviance reduction was not significant. Apart from "irrad670", all other predictors were significant in the final model, answering a key component of our research question that pressure ("pres"), depth ("depth"), temperature ("temp"), electrical conductivity ("cndc"), sea water salinity ("psal"), dissolved oxygen measure ("dox1"), dissolved organic matter ("cdom"), volume scattering function ("vbsc") and certain irradiation ("irrad443", "irrad490", "irrad555") all affect chlorophyll-a levels in the ocean. To confirm the P-value acceptance threshold used in our model, comparisons with the P-values in the linear model in [13] were made. While many of our predictors such as "temp", "psal" and "pres" were expectedly significant in agreement with multiple research papers on chlorophyll-a, there were also interesting variables such as vbsc which were not referenced in previous papers, suggesting it could be of interest for further studies.

The main factor in determining the GLM with an inverse link function as the final model resides in analysis of the Root Mean Square Error (RMSE), which

showed significantly better overall results when predicting the test datasets. Specifically, the RMSE for the final GLM on four different test data was 0.5399932, 0.5635647, 31.564871, 0.3053689. The third RMSE displays worrying results, highlighting the potential flaws of the model which will be explained in further detail below. Apart from that, the model shows extremely positive RMSE, in comparison to other similar linear models in [14] and [12] which had RMSE's of 0.3792097 and 9.94 respectively. Although it must be noted that RMSE is variable dependent and thus cannot directly be compared between models freely, this still conveys the validity of the GLM. The GLM with Gamma distribution and inverse link function has been a reasonable and justified approach to answering the research question, as it is able to highlight significant regressors that affect the levels of chlorophyll which is extremely useful for marine biologists and scientists in this field. However, the model is not without flaws. Firstly, due to the limited time to complete this report and the general lack of knowledge surrounding time-series data, a time-series model was not chosen. However, considering that the research question required building a predictive model, a time-series model could capture further complexity between chlorophyll-a and classic ocean properties and potentially resolve key issues in the GLM. The use of time-series data is also supported through research as a large majority of papers such as [12] either include time as a predictor variable in linear models or use time-series models. The acquisition of ocean nutrients data such as nitrate concentration has been shown to aid the predictive modelling of chlorophyll-a levels [15]. Combining the variables used in the final GLM model chosen with nutrients data could provide further insight into understanding the variation of chlorophyll-a levels in the ocean. Furthermore, another flaw in using the GLM for the ocean glider dataset is the prominence of chlorophyll-a values close to 0 (specifically under 0.01) which are unsuitable for Gamma distributions and thus reduces the validity of the model. Difficulties surrounding the zero values were met in the model selection process when stepwise feature selection was attempted with stepAIC function in R. Considering the importance of predicting values including and close to 0 in the ocean glider datasets for chlorophyll-a, a model that could handle both zero values and time-series would be a big improvement compared to the GLM. One such model that was used in the aforementioned research paper and also in [14] is the Wavelet Artificial Neural Network (WANN). The ANN architecture is a massive parallel distributed information-processing system that has certain performance characteristics resembling biological networks of the human brain, and the WANN is a hybrid version in which wavelet analysis is used as a data preprocessing technique to improve accuracy. In [14] , multiple methods of chlorophyll modelling and prediction including a linear model and WANN were presented, with the WANN being the most effective in terms of RMSE, R^2 and level of uncertainty. Although the WANN model would be a more impressive approach to the report, a general lack of understanding and experience in the field of neural networks made it unfeasible.

Chapter 7

Conclusion and Further Issues

Throughout the analysis, it has been confirmed that “depth”, “temp”, “cndc”, “psal”, “dox1”, “cdom”, “vbsc”, “irrad443”, “irrad490” and “irrad555” show relative significance in predicting chlorophyll-a levels. However further predictors such as specific nutrients and time could further help explain the complex variation in chlorophyll-a data. While the GLM addresses the predictive requirements for the research question, according to [14] there are models better suited for this study such as time-series models, of which the WANN model performed the best in terms of the standard criterion for prediction. Given more time and experience with neural networks, a WANN model for this report would be ideal and can be a point of further analysis in the future. In conclusion, according to this study the final GLM developed for chlorophyll-a prediction highlights significant predictors. The resulting significant predictors for chlorophyll-a levels can undergo further future research to gain valuable insights on the relevant fields. However, we recommend further research and consideration of new techniques and models such as WANN which better account for the data and can create more accurate predictions.

References

[1] N. G. National Geographic Society, Chlorophyll (Aug 2019).
URL https://www.nationalgeographic.org/encyclopedia/chlorophyll/

[2] E. C. Lamon III, A regression model for the prediction of chlorophyll a in lake okeechobee, florida, Lake and Reservoir Management 11 (4) (1995) 283–290.

[3] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[4] L. Breiman, Manual on setting up, using, and understanding random forests v3. 1, Statistics Department University of California Berkeley, CA, USA 1 (2002) 58.

[5] A. Liaw, M. Wiener, Classification and regression by randomforest, R News 2 (3) (2002) 18–22.
URL https://CRAN.R-project.org/doc/Rnews/

[6] Wikipedia contributors, Gamma distribution — Wikipedia, the free encyclopedia, [Online; accessed 01-November-2020] (2020).
URL https://en.wikipedia.org/w/index.php?title=Gamma_distribution&oldid=987121771

[7] M. N. Wright, A. Ziegler, ranger: A fast implementation of random forests for high dimensional data in c++ and r, Journal of Statistical Software 77 (1). doi:10.18637/jss.v077.i01.
URL http://dx.doi.org/10.18637/jss.v077.i01

[8] Wikipedia contributors, Decision tree learning — Wikipedia, the free encyclopedia, [Online; accessed 12-November-2020] (2020).
URL https://en.wikipedia.org/w/index.php?title=Decision_tree_learning&oldid=989356689

[9] N. M. F. S. NOAA, NASA, Usgcrp indicator details (2018).
URL https://www.globalchange.gov/browse/indicators/ocean-chlorophyll-concentrations

[10] A. Liaw, M. Wiener, Classification and regression by randomforest, Forest 23.

[11] D. Mwiti, Random forest regression: When does it fail and why? (May 2020).
URL https://neptune.ai/blog/random-forest-regression-when-does-it-fail-and~:text=Random%20forest%20is%20an%20ensemble%20of%20decision%20trees.&text=Each%20tree%20is%20created%20from,makes%20its%20own%20individual%20prediction

[12] T. Rajaee, A. Boroumand, Forecasting of chlorophyll-a concentrations in south san francisco bay using five different models, Applied Ocean Research 53 (2015) 208 – 217. doi:https://doi.org/10.1016/j.apor.2015.09.001.
URL http://www.sciencedirect.com/science/article/pii/S0141118715001157

[13] H. Çamdevýren, N. Demýr, A. Kanik, S. Keskýn, Use of principal component scores in multiple linear regression models for prediction of

chlorophyll-a in reservoirs, Ecological Modelling 181 (4) (2005) 581 – 589. doi:https://doi.org/10.1016/j.ecolmodel.2004.06.043.
URL http://www.sciencedirect.com/science/article/pii/S0304380004004004

[14] S. Shamshirband, E. J. Nodoushan, J. E. Adolf, A. A. Manaf, A. Mosavi, K. wing Chau, Ensemble models with uncertainty analysis for multi-day ahead forecasting of chlorophyll a concentration in coastal waters, Engineering Applications of Computational Fluid Mechanics 13 (1) (2019) 91–101. arXiv:https://doi.org/10.1080/19942060.2018.1553742, doi:10.1080/19942060.2018.1553742.
URL https://doi.org/10.1080/19942060.2018.1553742

[15] J. Marra, R. Bidigare, T. Dickey, Nutrients and mixing, chlorophyll and phytoplankton growth, Deep Sea Research Part A. Oceanographic Research Papers 37 (1) (1990) 127 – 143. doi:https://doi.org/10.1016/0198-0149(90)90032-Q.
URL http://www.sciencedirect.com/science/article/pii/019801499090032Q

[16] Y. Xie, J. Allaire, G. Grolemund, R Markdown, The Definitive Guide, Chapman and Hall/CRC, 2018.
URL https://bookdown.org/yihui/rmarkdown/

[17] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria (2017).
URL https://www.R-project.org/

[18] P. Lafaye de Micheaux, R. Drouilhet, B. Liquet, The R Software: Fundamentals of Programming and Statistical Analysis, Statistics and Computing, Springer New York, 2013.
URL https://books.google.fr/books?id=Ji-8BAAAQBAJ

[19] Wikipedia contributors, Random forest — Wikipedia, the free encyclopedia, [Online; accessed 10-November-2020] (2020).
URL https://en.wikipedia.org/w/index.php?title=Random_forest&oldid=985269509

[20] A. Cutler (Sep 2010).

[21] A. J. Irwin, Z. V. Finkel, Mining a sea of data: Deducing the environmental controls of ocean chlorophyll, PLOS ONE 3 (11) (2008) 1–6. doi:10.1371/journal.pone.0003836.
URL https://doi.org/10.1371/journal.pone.0003836

[22] S. Malek, S. M. S. Ahmad, S. K. K. Singh, P. Milow, A. Salleh, Assessment of predictive models for chlorophyll-a concentration of a tropical lake, in: BMC bioinformatics, Vol. 12, Springer, 2011, p. S12.

[23] H. Yajima, J. Derot, Application of the random forest model for chlorophyll-a forecasts in fresh and brackish water bodies in japan, using multivariate long-term databases, Journal of Hydroinformatics 20 (1) (2018) 206–220.

Appendix

Codes

7.0.1 GLM Code

```
# GLM Modelling
install.packages("DHARMa")
library(tidyverse)
library(magrittr)
library(readxl)
library(mgcv)
library(MASS)
library(SuppDists)
library(DHARMa)
library(boot)

dataset1 <- read.csv("Data/train1.csv")
dataset2_raw <- read.csv("Data/train2.csv")
dataset2 <- dataset2_raw[-c(3,4)] # Get rid of lat and long

# Merge datasets together
merged <- rbind(dataset1, dataset2)

# Remove rows with missing values
model_data <- na.omit(merged)
attach(model_data)

# Create a running average vector of all variables - takes average of every 5
cphl_avg5 <- rep(NA, 308835)
depth_avg5 <- rep(NA, 308835)
temp_avg5 <- rep(NA, 308835)
cndc_avg5 <- rep(NA, 308835)
psal_avg5 <- rep(NA, 308835)
dox1_avg5 <- rep(NA, 308835)
cdom_avg5 <- rep(NA, 308835)
vbsc_avg5 <- rep(NA, 308835)
irrad443_avg5 <- rep(NA, 308835)
irrad490_avg5 <- rep(NA, 308835)
irrad555_avg5 <- rep(NA, 308835)
irrad670_avg5 <- rep(NA, 308835)
```

```
j = 1

# Take average of every 5 data points in train set
for (i in seq(from=1, to=1544171, by=5)) {
  cphl_avg5[j] = mean(cphl[i:(i+4)])
  depth_avg5[j] = mean(depth[i:(i+4)])
  temp_avg5[j] = mean(temp[i:(i+4)])
  cndc_avg5[j] = mean(cndc[i:(i+4)])
  psal_avg5[j] = mean(psal[i:(i+4)])
  dox1_avg5[j] = mean(dox1[i:(i+4)])
  cdom_avg5[j] = mean(cdom[i:(i+4)])
  vbsc_avg5[j] = mean(vbsc[i:(i+4)])
  irrad443_avg5[j] = mean(irrad443[i:(i+4)])
  irrad490_avg5[j] = mean(irrad490[i:(i+4)])
  irrad555_avg5[j] = mean(irrad555[i:(i+4)])
  irrad670_avg5[j] = mean(irrad670[i:(i+4)])
  j = j + 1
}

# Combine all averaged columns into 1 dataframe
model_data_avg5 <- data.frame(cbind(cphl_avg5,depth_avg5,temp_avg5,cndc_avg5,ps
                                    cdom_avg5,vbsc_avg5,irrad443_avg5,irrad490_
                                    irrad555_avg5,irrad670_avg5))

# Only take rows with positive cphl_avg values
model_data_avg5_pos <- model_data_avg5[model_data_avg5$cphl_avg5 > 0.01,]
#Rename columns
names(model_data_avg5_pos)[1] <- "cphl"
names(model_data_avg5_pos)[2] <- "depth"
names(model_data_avg5_pos)[3] <- "temp"
names(model_data_avg5_pos)[4] <- "cndc"
names(model_data_avg5_pos)[5] <- "psal"
names(model_data_avg5_pos)[6] <- "dox1"
names(model_data_avg5_pos)[7] <- "cdom"
names(model_data_avg5_pos)[8] <- "vbsc"
names(model_data_avg5_pos)[9] <- "irrad443"
names(model_data_avg5_pos)[10] <- "irrad490"
names(model_data_avg5_pos)[11] <- "irrad555"
names(model_data_avg5_pos)[12] <- "irrad670"




# Method of Moments Estimation of k = shape and theta = scale parameters for
```

```r
sample_mean <- mean(model_data_avg5_pos$cphl)
sample_var <- var(model_data_avg5_pos$cphl)

k <- (sample_mean)^2/sample_var
theta <- sample_var/sample_mean

# histogram of cphl-average5 combined with density curve for inverse gaussian
par(mfrow=c(1,1))
hist(cphl, probability = TRUE, xlab = "Chlorophyll-a Concentration", main = "Hi
# Method of Moments estimate of gamma distribution
curve(dgamma(x,shape = k, scale = theta), add = TRUE, col="red")



##############################################
##############################################
##############################################
##############################################
##############################################
attach(model_data_avg5_pos)

fit1 <- glm(cphl~., family = Gamma(link = "inverse"), data = model_data_avg5_po
            start = rep(1,12))
fit2 <- glm(cphl~1, start = rep(1,1), family = Gamma(link = "inverse"), data =

fit3 <- glm(cphl~depth, start = rep(1,2), family = Gamma(link = "inverse"), da
fit3.1 <- glm(cphl~depth+temp, start = rep(1,3), family = Gamma(link = "inverse
fit3.2 <- glm(cphl~depth+psal, start = rep(1,3), family = Gamma(link = "inverse
fit3.3 <- glm(cphl~depth+dox1, start = rep(1,3), family = Gamma(link = "inverse
fit3.4 <- glm(cphl~depth+cdom, start = rep(1,3), family = Gamma(link = "inverse
fit3.5 <- glm(cphl~depth+vbsc, start = rep(1,3), family = Gamma(link = "inverse
fit3.6 <- glm(cphl~depth+irrad443, start = rep(1,3), family = Gamma(link = "inv
fit3.7 <- glm(cphl~depth+irrad490, start = rep(1,3), family = Gamma(link = "inv
fit3.8 <- glm(cphl~depth+irrad555, start = rep(1,3), family = Gamma(link = "inv
fit3.9 <- glm(cphl~depth+irrad670, start = rep(1,3), family = Gamma(link = "inv

aic3.1 <- AIC(fit3.1)
aic3.2 <- AIC(fit3.2)
aic3.3 <- AIC(fit3.3)
aic3.4 <- AIC(fit3.4)
aic3.5 <- AIC(fit3.5)
aic3.6 <- AIC(fit3.6) ### WINNER
aic3.7 <- AIC(fit3.7)
aic3.8 <- AIC(fit3.8)
aic3.9 <- AIC(fit3.9)
```

```
min3 <- min(c(aic3.1,aic3.2,aic3.3,aic3.4,aic3.5,aic3.6,aic3.7,aic3.8,aic3.9))
(min3 == c(aic3.1,aic3.2,aic3.3,aic3.4,aic3.5,aic3.6,aic3.7,aic3.8,aic3.9))

fit4 <- glm(cphl~depth+irrad443, start = rep(1,3), family = Gamma(link = "inve
fit4.1 <- glm(cphl~depth+irrad443+temp, start = rep(1,4), family = Gamma(link =
fit4.2 <- glm(cphl~depth+irrad443+psal, start = rep(1,4), family = Gamma(link =
fit4.3 <- glm(cphl~depth+irrad443+dox1, start = rep(1,4), family = Gamma(link =
fit4.4 <- glm(cphl~depth+irrad443+cdom, start = rep(1,4), family = Gamma(link =
fit4.5 <- glm(cphl~depth+irrad443+vbsc, start = rep(1,4), family = Gamma(link =
fit4.6 <- glm(cphl~depth+irrad443+irrad490, start = rep(1,4), family = Gamma(li
fit4.7 <- glm(cphl~depth+irrad443+irrad555, start = rep(1,4), family = Gamma(li
fit4.8 <- glm(cphl~depth+irrad443+irrad670, start = rep(1,4), family = Gamma(li

aic4.1 <- AIC(fit4.1)
aic4.2 <- AIC(fit4.2)
aic4.3 <- AIC(fit4.3)
aic4.4 <- AIC(fit4.4) ### WINNER
aic4.5 <- AIC(fit4.5)
aic4.6 <- AIC(fit4.6)
aic4.7 <- AIC(fit4.7)
aic4.8 <- AIC(fit4.8)

min4 <- min(c(aic4.1,aic4.2,aic4.3,aic4.4,aic4.5,aic4.6,aic4.7,aic4.8))
(min4 == c(aic4.1,aic4.2,aic4.3,aic4.4,aic4.5,aic4.6,aic4.7,aic4.8))

fit5 <- glm(cphl~depth+irrad443+cdom, start = rep(1,4), family = Gamma(link =
fit5.1 <- glm(cphl~depth+irrad443+cdom+temp, start = rep(1,5), family = Gamma(
fit5.2 <- glm(cphl~depth+irrad443+cdom+psal, start = rep(1,5), family = Gamma(
fit5.3 <- glm(cphl~depth+irrad443+cdom+dox1, start = rep(1,5), family = Gamma(
fit5.4 <- glm(cphl~depth+irrad443+cdom+vbsc, start = rep(1,5), family = Gamma(
fit5.5 <- glm(cphl~depth+irrad443+cdom+irrad490, start = rep(1,5), family = Ga
fit5.6 <- glm(cphl~depth+irrad443+cdom+irrad555, start = rep(1,5), family = Ga
fit5.7 <- glm(cphl~depth+irrad443+cdom+irrad670, start = rep(1,5), family = Ga

aic5.1 <- AIC(fit5.1) ### WINNER
aic5.2 <- AIC(fit5.2)
aic5.3 <- AIC(fit5.3)
aic5.4 <- AIC(fit5.4)
aic5.5 <- AIC(fit5.5)
aic5.6 <- AIC(fit5.6)
aic5.7 <- AIC(fit5.7)

min5 <- min(c(aic5.1,aic5.2,aic5.3,aic5.4,aic5.5,aic5.6,aic5.7))
(min5 == c(aic5.1,aic5.2,aic5.3,aic5.4,aic5.5,aic5.6,aic5.7))

fit6 <- glm(cphl~depth+irrad443+cdom+temp, start = rep(1,5), family = Gamma(li
```

```
fit6.1 <- glm(cphl~depth+irrad443+cdom+temp+psal, start = rep(1,6), family = Ga
fit6.2 <- glm(cphl~depth+irrad443+cdom+temp+dox1, start = rep(1,6), family = Ga
fit6.3 <- glm(cphl~depth+irrad443+cdom+temp+vbsc, start = rep(1,6), family = Ga
fit6.4 <- glm(cphl~depth+irrad443+cdom+temp+irrad490, start = rep(1,6), family
fit6.5 <- glm(cphl~depth+irrad443+cdom+temp+irrad555, start = rep(1,6), family
fit6.6 <- glm(cphl~depth+irrad443+cdom+temp+irrad670, start = rep(1,6), family

aic6.1 <- AIC(fit6.1) ### WINNER
aic6.2 <- AIC(fit6.2)
aic6.3 <- AIC(fit6.3)
aic6.4 <- AIC(fit6.4)
aic6.5 <- AIC(fit6.5)
aic6.6 <- AIC(fit6.6)


min6 <- min(c(aic6.1,aic6.2,aic6.3,aic6.4,aic6.5,aic6.6))
(min6 == c(aic6.1,aic6.2,aic6.3,aic6.4,aic6.5,aic6.6))

fit7 <- glm(cphl~depth+irrad443+cdom+temp+psal, start = rep(1,6), family = Gamm
fit7.1 <- glm(cphl~depth+irrad443+cdom+temp+psal+dox1, start = rep(1,7), family
fit7.2 <- glm(cphl~depth+irrad443+cdom+temp+psal+vbsc, start = rep(1,7), family
fit7.3 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490, start = rep(1,7), fa
fit7.4 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad555, start = rep(1,7), fa
fit7.5 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad670, start = rep(1,7), fa

aic7.1 <- AIC(fit7.1)
aic7.2 <- AIC(fit7.2)
aic7.3 <- AIC(fit7.3) ### WINNER
aic7.4 <- AIC(fit7.4)
aic7.5 <- AIC(fit7.5)



min7 <- min(c(aic7.1,aic7.2,aic7.3,aic7.4,aic7.5))
(min7 == c(aic7.1,aic7.2,aic7.3,aic7.4,aic7.5))

fit8 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490, start = rep(1,7), fami
fit8.1 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+dox1, start = rep(1,8
fit8.2 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+vbsc, start = rep(1,8
fit8.3 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad555, start = rep
fit8.4 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad670, start = rep

aic8.1 <- AIC(fit8.1)
aic8.2 <- AIC(fit8.2)
aic8.3 <- AIC(fit8.3) ### WINNER
aic8.4 <- AIC(fit8.4)
```

```
min8 <- min(c(aic8.1,aic8.2,aic8.3,aic8.4))
(min8 == c(aic8.1,aic8.2,aic8.3,aic8.4))

fit9 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad555, start = rep(
fit9.1 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad555+irrad490+dox1, start
fit9.2 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad555+vbsc, start
fit9.3 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad555+irrad670, s

aic9.1 <- AIC(fit9.1) ### WINNER
aic9.2 <- AIC(fit9.2)
aic9.3 <- AIC(fit9.3)

min9 <- min(c(aic9.1,aic9.2,aic9.3))
(min9 == c(aic9.1,aic9.2,aic9.3))

fit10 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad555+irrad490+dox1, start =
fit10.1 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad555+dox1+vbsc,
fit10.2 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad555+dox1+irrad

aic10.1 <- AIC(fit10.1) ### WINNER
aic10.2 <- AIC(fit10.2)

min10 <- min(c(aic10.1,aic10.2))
(min10 == c(aic10.1,aic10.2))

fit11 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad555+dox1+vbsc, s
fit11.1 <- glm(cphl~depth+irrad443+cdom+temp+psal+irrad490+irrad555+dox1+vbsc+

min11 <- AIC(fit11.1)


ultimate_min_AIC <- min(c(min3,min4,min5,min6,min7,min8,min9,min10,min11))
(ultimate_min_AIC == c(min3,min4,min5,min6,min7,min8,min9,min10,min11))


####
# Final Models that we compare for AIC and RMSE
####
gamma_invsmall <- fit11.1
gamma_log <- glm(cphl~., family = Gamma(link = "log"), data = model_data_avg5_
gamma_invfull <- glm(cphl~., family=Gamma(link = "inverse"),
                     data = model_data_avg5_pos, start = rep(1,12))
gamma_invno670 <- glm(cphl~.-irrad670, family=Gamma(link = "inverse"),
                      data = model_data_avg5_pos, start = rep(1,11))
```

```r
final_model <- gamma_invfull

#### GLM Diag Plots ### Library(boot)

simout<- simulateResiduals(final_model, n=250)
plot(simout)

#testing glm1 diagnostics
#### 2 Lines Below take around 30 minutes to run with Windows 10 - 64 bit, in
# glm1.diag<-glm.diag(glm1)
# glm.diag.plots(glm1, glm1.diag)




##################################################
###Load Test Sets
test1 <- read.csv("Data/test_autumn.csv")
test2 <- read.csv("Data/test_spring.csv")
test3 <- read.csv("Data/test_summer.csv")
test4 <- read.csv("Data/test_winter.csv")

### RMSE on Gamma Inverse (no cndc)
a1 <- test1$cphl
a1.pred <- predict(gamma_invno670, newdata = test1, type = "response")
sum(is.na(a1.pred))
rmse1_no670 <- sqrt(mean((a1-a1.pred)^2))

a2 <- test2$cphl
a2.pred <- predict(gamma_invno670, newdata = test2, type = "response")
sum(is.na(a2.pred))
rmse2_no670 <- sqrt(mean((a2-a2.pred)^2))

a3 <- test3$cphl
a3.pred <- predict(gamma_invno670, newdata = test3, type = "response")
sum(is.na(a3.pred))
rmse3_no670 <- sqrt(mean((a3-a3.pred)^2))

a4 <- test4$cphl
a4.pred <- predict(gamma_invno670, newdata = test4, type = "response")
sum(is.na(a4.pred))
rmse4_no670 <- sqrt(mean((a4-a4.pred)^2))
gamma_invno670.rmse <- c(rmse1_no670,rmse2_no670,rmse3_no670,rmse4_no670)

### RMSE on Gamma Inverse (no cndc)
```

```r
y1 <- test1$cphl
y1.pred <- predict(gamma_invsmall, newdata = test1, type = "response")
sum(is.na(y1.pred))
rmse1 <- sqrt(mean((y1-y1.pred)^2))

y2 <- test2$cphl
y2.pred <- predict(gamma_invsmall, newdata = test2, type = "response")
sum(is.na(y2.pred))
rmse2 <- sqrt(mean((y2-y2.pred)^2))

y3 <- test3$cphl
y3.pred <- predict(gamma_invsmall, newdata = test3, type = "response")
sum(is.na(y3.pred))
rmse3 <- sqrt(mean((y3-y3.pred)^2))

y4 <- test4$cphl
y4.pred <- predict(gamma_invsmall, newdata = test4, type = "response")
sum(is.na(y4.pred))
rmse4 <- sqrt(mean((y4-y4.pred)^2))
gamma_invsmall.rmse <- c(rmse1,rmse2,rmse3,rmse4)

### RMSE on Gamma Inverse FULL
z1 <- test1$cphl
z1.pred <- predict(gamma_invfull, newdata = test1, type = "response")
sum(is.na(z1.pred))
rmse1_invF <- sqrt(mean((z1-z1.pred)^2))

z2 <- test2$cphl
z2.pred <- predict(gamma_invfull, newdata = test2, type = "response")
sum(is.na(z2.pred))
rmse2_invF <- sqrt(mean((z2-z2.pred)^2))

z3 <- test3$cphl
z3.pred <- predict(gamma_invfull, newdata = test3, type = "response")
sum(is.na(z3.pred))
rmse3_invF <- sqrt(mean((z3-z3.pred)^2))

z4 <- test4$cphl
z4.pred <- predict(gamma_invfull, newdata = test4, type = "response")
sum(is.na(z4.pred))
rmse4_invF <- sqrt(mean((z4-z4.pred)^2))
gamma_invfull.rmse <- c(rmse1_invF,rmse2_invF,rmse3_invF,rmse4_invF)

### RMSE on Gamma Log
x1 <- test1$cphl
x1.pred <- predict(gamma_log, newdata = test1, type = "response")
```

```
sum(is.na(x1.pred))
rmse1_log <- sqrt(mean((x1-x1.pred)^2))

x2 <- test2$cphl
x2.pred <- predict(gamma_log, newdata = test2, type = "response")
sum(is.na(x2.pred))
rmse2_log <- sqrt(mean((x2-x2.pred)^2))

x3 <- test3$cphl
x3.pred <- predict(gamma_log, newdata = test3, type = "response")
sum(is.na(x3.pred))
rmse3_log <- sqrt(mean((x3-x3.pred)^2))

x4 <- test4$cphl
x4.pred <- predict(gamma_log, newdata = test4, type = "response")
sum(is.na(x4.pred))
rmse4_log <- sqrt(mean((x4-x4.pred)^2))
gamma_log.rmse <- c(rmse1_log,rmse2_log,rmse3_log,rmse4_log)

# Compare all
gamma_invfull.rmse ###### Winner
gamma_invno670.rmse
gamma_invsmall.rmse
gamma_log.rmse
```

7.0.2 Ranger RandomForest Implementation

```
install.packages('rsample')
install.packages('randomForest')
install.packages('ranger')
install.packages('caret')
install.packages('h2o')
library(rsample)      # data splitting
library(randomForest) # basic implementation
library(ranger)       # a faster implementation of randomForest
library(caret)        # an aggregator package for performing many machine lea
library(h2o)
library(dplyr)
library(caret)
library(ranger)

model_data1 <- filter(model_data,cphl_flag == 1 & pres_flag == 1 & depth_flag =
                        psal_flag == 1 & dox1_flag == 1 & cdom_flag == 1 & vbs
                        irrad443_flag == 1 & irrad490_flag == 1 & irrad555_flag
#removing rows with missing data, dropping variables that don't contribute
model_data <- na.omit(model_data1)
```

```r
model_data <- model_data1[-c(2,3,4,18,19,20,21,22,23,24,25,26,27,28,29,30,31,3

m1 <- model_data[1:389675,]


m1_fit <- ranger(cphl ~ ., data = m1)
#parameters (num.trees = 500,max.depth = 8,probability = TRUE)
m1_fit$num.trees
m1_fit$mtry
#Overall out of bag prediction error (prediciton.error), for regression uses
m1_fit$prediction.error
m1_fit$r.squared
```

```r
ss <- model_data1[sample(nrow(model_data1), 8523), ]
sample_size = floor(0.75*nrow(ss))
sample_size
set.seed(123)
train_ind = sample(seq_len(nrow(model_data1)),size = sample_size)
train = model_data1[train_ind,]
test = model_data1[-train_ind,]

m1 <- randomForest(formula = cphl~., data = train)
predict(m1,data=test)
plot(m1)
which.min(m1$mse)
```

For full code please visit:
https://github.com/kurtwang8/Team-8---Oceanography-modelling-phytoplankton-

Tables

7.0.3 Reference Table for Variable abbreviations

Variable	Description
cphl	Chlorophyll-a
pres	Pressure
depth	Depth
temp	Temperature
cndc	Electrical Conductivity
psal	Sea Water Salinity
dox1	Dissolved Molecular Oxygen
cdom	Coloured Dissolved Organic Matter
vbsc	Volume Scattering Function
bbp	Particle Backscattering coefficient
irrad443	Downwelling Spectral Irradiance in Sea Water - wavelength 443
irrad490	Downwelling Spectral Irradiance in Sea Water - wavelength 490
irrad555	Downwelling Spectral Irradiance in Sea Water - wavelength 555

Variable	Description
irrad670	Downwelling Spectral Irradiance in Sea Water - wavelength 670