STAT 450 Group Report

The Effects of Climate Variables on Average Stream Flow for Canadian Watersheds

By Anjali Chauhan (57181489), Kelvin Li (82433020),
Kohl Peterson (69586204), Vanessa Bayubaskoro (45602166)

March 26, 2022

1. Summary

In this analysis, we sought to develop a model that predicts the annual average streamflow for Canadian watersheds by studying the effects of climate variables on the annual average streamflow to provide crucial information for efficient water resource management. This helps in reducing the financial loss due to floods, droughts and dam mismanagement. By training an ensemble of prediction models like XGBoost, Gradient Boosting Machine, Random Forest, Quantile Regression Forests, Support Vector Machine and Linear Regression with important variables selected through variable importance techniques like Boruta and Forward Stepwise Regression and predicting the annual average streamflow, we achieved a high prediction performance with the lowest Root Mean Square Error (RMSE) of 0.1878 for the Ensemble Model. These techniques with further analysis can be broadly applied to predict average annual streamflow across different watersheds throughout the planet without the spatial and temporal input.

2. Introduction

A watershed refers to an area of land where rainfall and snowmelt collect and stream into a common outlet like rivers, lakes and other bodies of water. Understanding watershed stream flow is important for water resource management e.g. irrigation, hydroelectric power and flood control. The study aims to investigate and understand the effect of climate variables on the watershed's streamflow.
The analysis aims to address the following questions:

- Can the data from one catchment be used to extrapolate stream flow in another catchment given the climate variables?
- Can we detect the unusual streamflow activity accurately that can lead to severe adverse effects for the nearby ecosystem and populated areas?

More specifically, the analysis has the following objectives to answer the above questions:

- To build an outlier detection system to detect the unusual streamflow activity (extreme values)
- To translate the relationship between the response and multiple predictors, measured over many years, into insightful visualisations
- To find important climate variable(s) and determine the effect of said variable(s) on the streamflow
- To model and predict the average stream flow values.

This report summarises all of the primary statistical modelling and analysis results associated with the study. The remainder of this report is organised as follows: Section 3 describes the data collection, provides measurement of the variables and summarises the data. Section 4 presents the data preprocessing and statistical modelling techniques used to answer the client's research questions. Section 5 summarises and interprets the results of the statistical analysis conducted. Appendices are provided for further exploratory data analysis along with the code used for the statistical modelling. Section 6 describes the outlier detection performed on the predicted stream flow values. Lastly, Section 7 presents the limitations and challenges in conducting this analysis.

3. Data

3.1 Description

The data was collected daily by satellites and an aggregation of the data (annual averages) was provided for the purpose of this analysis. Data contains observations from 23 medium-sized water catchment areas located around Canada. The size of these watershed areas ranges from 50 km^2 to 10,000 km^2. The data of various climate variables was taken from the year 1980 to 2018. There are 774 rows and 12 columns with no missing data. An additional dataset with the watershed shape data (Longitude, Latitude) was also provided by the client for further analysis of the effect of spatial features on the streamflow.

Table 1: Description of Variables Used for Analysis

	Variable	Abbrev.	Unit	Description
1.	***(Response)*** Annual Average Streamflow	Q	$m^3\ year^{-1}$	The average daily stream flow recorded for a year
2.	Mean Yearly Potential Evaporation	PEVA	ml	The amount of evaporation that would occur if a sufficient water source were available
3.	Mean Snow Density	SDEN	kg^3m^{-3}	The density of the snow
4.	Mean Snow Depth	SDEP	m	The depth of new and old snow that remains on the ground at observation time
5.	Mean Snow Depth of Snow Water Equivalent	SWEQ	m	Water equivalent of melted snow collected in the gauge since the last observation
6.	Mean Yearly Temperature	TEMP	$°C$	The temperature measured in °C
7.	Mean Yearly Snowfall	SFAL	m	The record of snowfall (snow, ice pellets) since the previous snowfall observation (24 hours)
8.	Mean Yearly Snowmelt	SMELT	m	The depth of runoff produced from melting snow
9.	Mean Yearly Total Precipitation	PREC	m	The depth of total rainfall and water-equivalent of snowfall
10.	Year	-	year	The year the data was recorded in
11.	Grid Code	-	-	Grid code where the watershed is located
12.	Mean Yearly Evaporation	EVA	m	Depth of water evaporates from the catchment area

Table 2: Summary Statistics of All Climate Variables

Var	EVA	PEVA	SDEN	SDEP	SWEQ	SFAL	SMELT	TEMP	PREC	Q
Min	0.123	71.46	131.1	0.048	0.008	35.61	17.78	-7.153	0.364	0.045
25%	0.293	1140.49	166.9	0.213	0.044	96.80	83.57	-1.417	0.684	0.599
50%	0.352	1352.98	194.5	0.319	0.077	128.13	117.0	0.190	0.768	0.899
Mean	0.364	1238.07	195.0	0.376	0.097	136.60	127.2	0.135	0.779	0.949

Var	EVA	PEVA	SDEN	SDEP	SWEQ	SFAL	SMELT	TEMP	PREC	Q
75%	0.427	1509.39	217.5	0.507	0.133	166.81	163.3	1.806	0.872	1.233
Max	0.712	1999.25	290.7	1.120	0.382	332.46	333.8	7.014	1.270	2.436

3.2 Exploratory Data Analysis

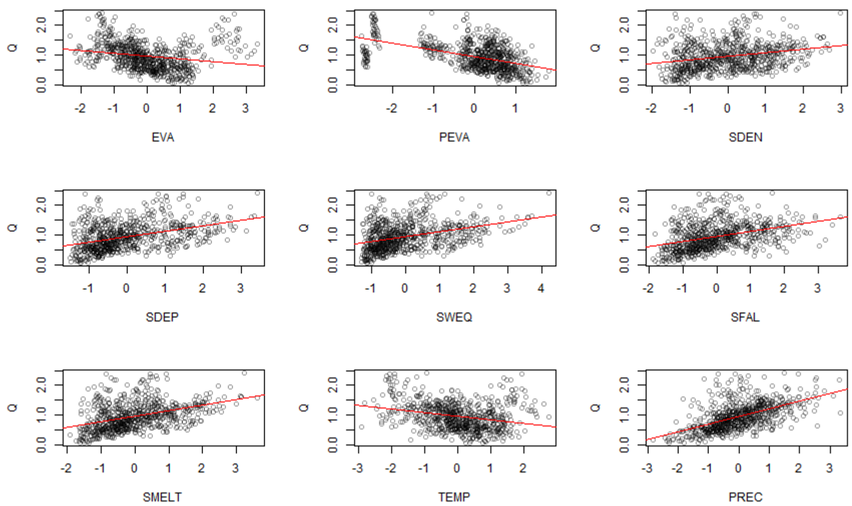

Fig. 1: Relationship between Climate Variables and Streamflow

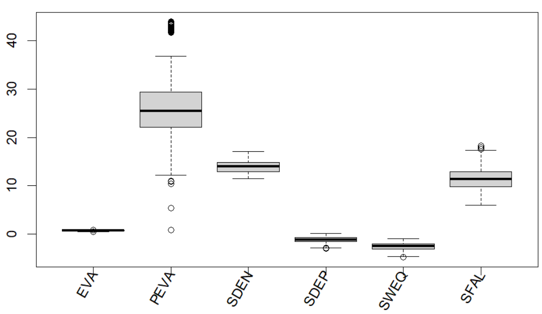

Fig. 2: Before Scaling

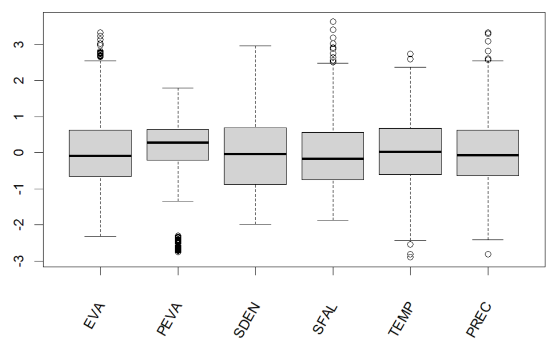

Fig. 3: After Scaling

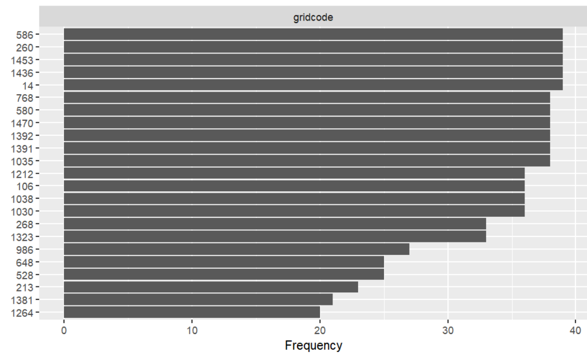

Fig. 4: Numbers of Observation by Gridcode

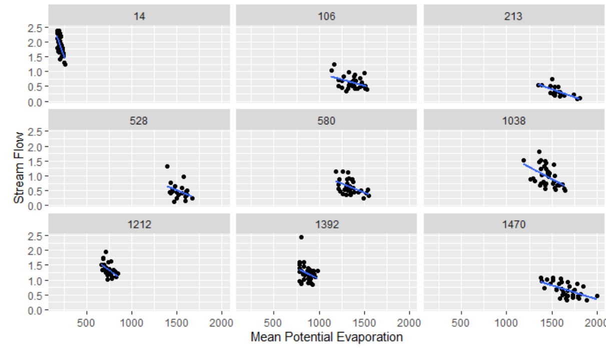

Fig. 5: Stream Flow vs Mean Potential Evaporation for Selected Gridcodes

Looking more closely at various variables broken out by `gridcode` (*see Fig. 5 as an example*), some interesting results were seen. For many `gridcode`'s, there appears to be a much stronger linear relationship between climate variables and the stream flow and these relationships appear to have different intercept and slope values. This suggests that there may not be a one-size-fits-all approach to fitting a regression model based on annual climate variables alone and different slopes for different `gridcode`'s may need to be considered for further analysis in the future.

4. Methods

4.0. Pipeline

Below in Fig. 6, we have a Proof-of-Concept pipeline that addresses all of the client's research questions. A breakdown of each of the steps shown in the end-to-end workflow diagram is covered below.

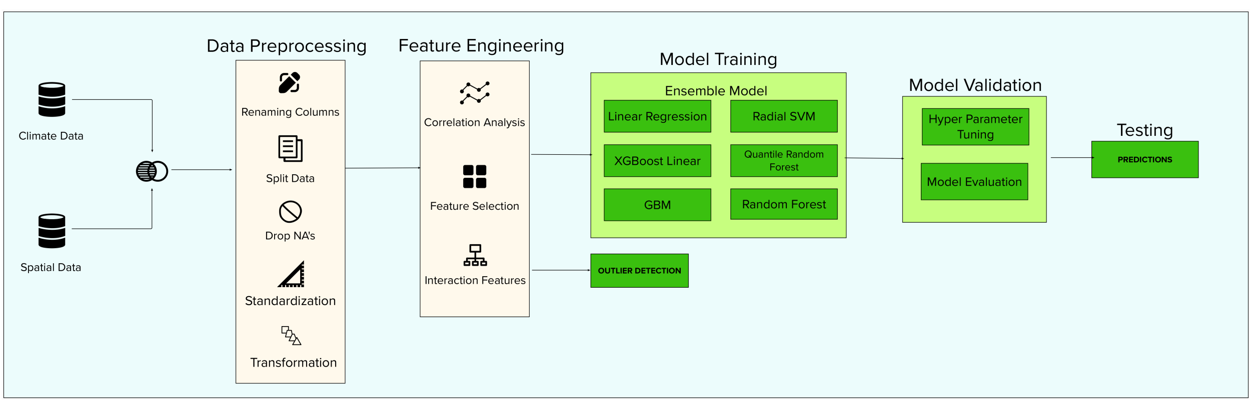

Fig. 6: End-to-End Pipeline

4.1 Data Pre-Processing

As part of the data pre-processing pipeline, several different transformations were performed in order to ensure consistency in data and to eliminate skewness and ensuring an approximately normal distribution of the explanatory variables. As this is one of the assumptions of our Linear Model that needs to be satisfied.

4.1.1 Feature Transformations

In addition to Z-Score Standardization, which refers to scaling and centering of the distribution to ensure the data is not on a varying scale and is internally consistent, several feature transformations were performed to deal with both left and right skewed feature distributions. Table 3 shows all the transformations performed in order to achieve the desired results which are highlighted by Fig. 7 and Fig. 8.

Table 3: Feature Transformations

Variable	Description	Transformation	Skewness
`EVA`	Mean Yearly Evaporation	Cube Root Transform	Right Skewed
`SDEN`	Mean Snow Density	Square Root Transform	Right Skewed
`SDEP`	Mean Snow Depth	Log Transform	Right Skewed
`SWEQ`	Mean Snow Depth of Snow Water Equivalent	Log Transform	Right Skewed
`SFAL`	Mean Yearly Snowfall	Square Root Transform	Right Skewed
`SMELT`	Mean Yearly Snowmelt	Square Root Transform	Right Skewed
`PEVA`	Mean Yearly Potential Evaporation	Shifted Square Root Transform	Left Skewed

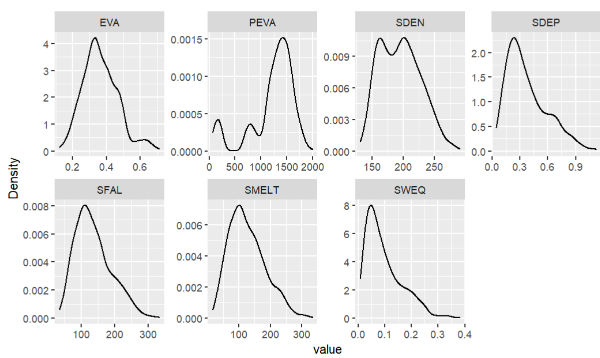

Fig. 7: Before Transformation

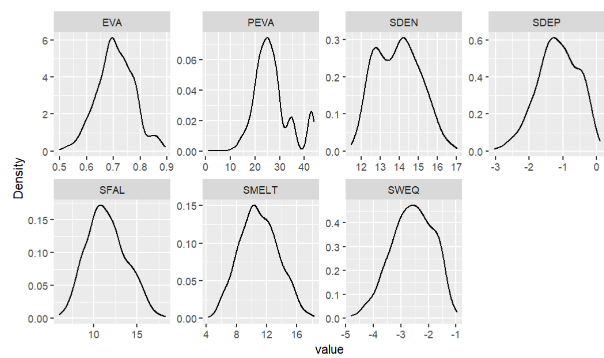

Fig. 8: After Transformation

4.2 Feature Selection

One of the most important step in our pipeline was Feature Selection as it is one of the main objectives. We have used several different methods in order to achieve this objective:

4.2.1. Pairwise Correlation Analysis

Before using any of the traditional feature selection techniques mentioned below, we investigated if any of the features were highly correlated. We see from the Correlation Matrix below that the Mean Snow Depth, Mean Snow Depth of Snow Water Equivalent and the Mean Yearly Snow Melt are highly correlated.

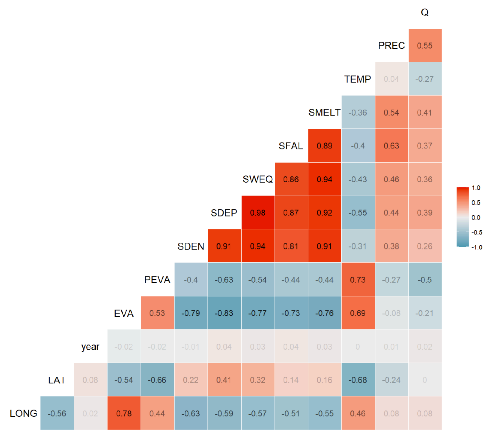

Fig. 9: Heatmap Presenting Correlation Between Variables

4.2.2 Principal Component Analysis

For our initial MVP, we have dropped the highly correlated features stated above as there is insufficient information contained in the linear combination of the above features leading to redundancy. To further deal with this issue, we have also performed PCA as part of our exploratory analysis. The interpretation of these principal components is based on finding which variables are most strongly correlated with each component, i.e., which of these features have a large absolute relative importance, the farthest from zero in either direction. The importance of each feature is reflected by the magnitude of the corresponding values in the eigenvectors.

The first principal component is strongly correlated with three of the original variables (SDEP, SWEQ, SMELT). The first principal component increases with increasing eithr one fo these highly correlated features. If one increases, then the remaining ones tend to increase as well. Furthermore, we see that the first principal component correlates most strongly with the Mean Snow Depth (SDEP). It means that gridcodes with high values of stream flow values tend to have a lot of a high Mean Snow Depth. Whereas gridcodes with small values would have very low SDEP.

One caveat here is given the data is not high dimensional and PCA is commonly used for dimensionality reduction which is why these principal components are not being used as model input because we will only be providing the models with two inputs. Thus, we are considered to choose the set of the original parameters that describe a large amount of the variation.

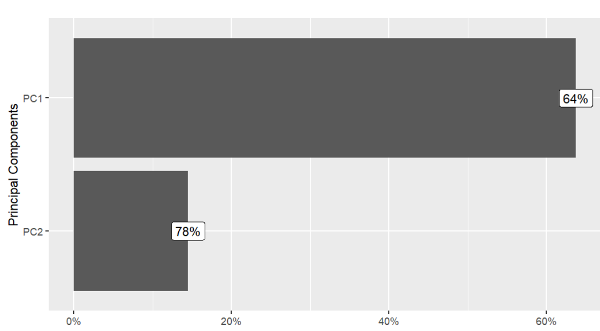

Fig. 10: Percentage of Variance Explained by Principal Components

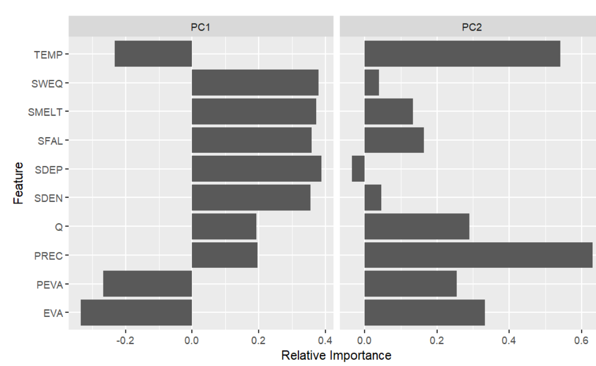

Fig. 11: Feature by Relative Importance in Different Principal Components

4.2.2. Variable Importance using `boruta` package

From Fig. 12, we see that the most important variables selected ranked in the following order: PRECIP, gridcode, PEVA, EVA, TEMP, SFAL, SDEN.

This method is built on a random forest classifier. It ranks features based on their importance measure i.e. Mean Decrease Accuracy (MDA) where higher means more important. MDA measures how much accuracy the model losses by excluding each variable. The more the accuracy degrades, the more important the variable is.

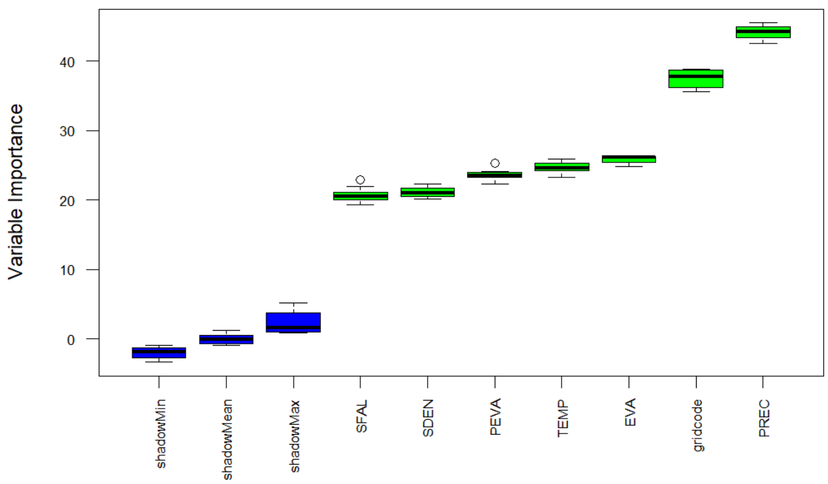

Fig 12: Variable Importance (using 'boruta' package)

4.2.3. Forward Stepwise Regression

This method was used to find important variables as well but it was exclusively used to find the important interaction *(please refer to A.1.1 for the definition)* terms for the Linear models. The two plots below are comparing the performance metrics of the linear model with different subset sizes of the feature space (with/without interaction terms respectively). We see that after 7 features without any interaction terms, the curve for all metrics plateaus indicating that 7 features is a good subset size to maximize the performance. Similarly, when including the interaction terms, we see the same result after 20 variables. *(The complete list of variables can be found in the Appendix I (A.1.2)).*

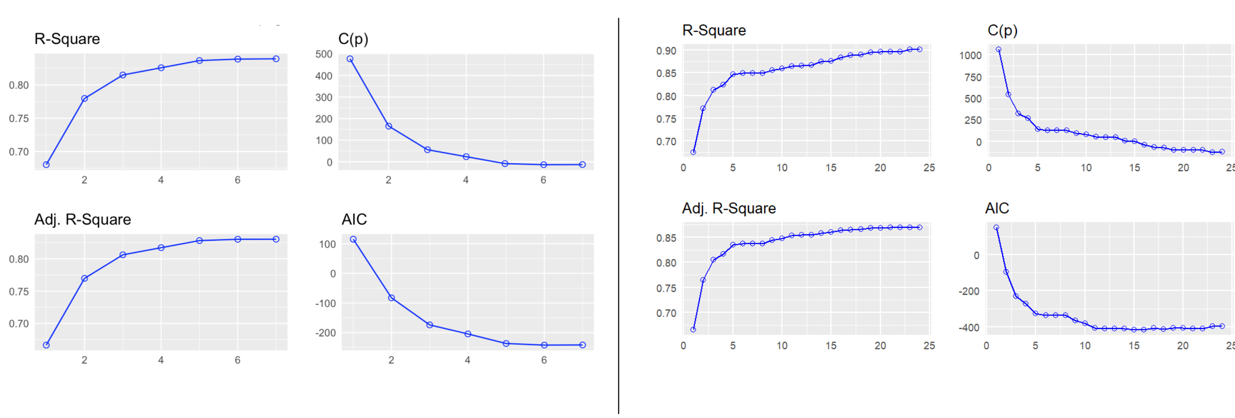

Fig 13: Feature Selection Without Interaction Terms (Forward Stepwise Regression)

Fig 14: Feature Selection With Interaction Terms (Forward Stepwise Regression)

4.3. Model Training and Validation

A good variety of models were implemented as a part of our analysis for extensive results. Two sets of models where trained to capture both the effect of individual co-variate terms and the interaction terms on the predictive performance of the model. These models were cross-validated (10-fold, Repeated CV) and their hyper parameters were fine tuned using Random Search. Please refer to Table 4 and Table 5 for results that denote the predictive performance of the models.

4.3.1. Linear Models

After feature selection, we trained a Linear Regression model with a 10-fold cross validation to test the model performance on the training data. We train a similar set of models with interaction terms as features and we see slightly better results than the former method.

4.3.2. Support Vector Machines

To add further complexities to the previous models, we trained Support Vector Machines that expanded our feature space using different kernels. We have a radial kernel to compare performance of models without the complexities relating to Linearity respectively. In the radial kernel, only the neighboring behaviour of data is taken into account which means only those data points influence the modelling compared to the Linear SVM whose performance is similar to a Linear model. From Table 4 and Table 5, we see that the Radial SVM without interaction terms performs slightly better than Radial SVM with the interaction terms.

4.3.3. Tree Models

We used three tree-based models such as Random Forest, XGBoost, Quantile Regression Forest (QRF) and Gradient Boosting Machine to improve the performance compared to the above models. From Table 4 and Table 5, we see that the Random Forest, QRF and XGBoost with interaction terms has better results compared to without interaction terms. However, the GBM without interaction terms gives slightly better results than the GBM with interaction terms. The best tree model is the Quantile Regression Forest Model with the lowest RMSE of 0.227 (with interaction terms).

4.3.4. Ensemble Models

We trained two Ensemble models with and without the interaction terms that combine the above listed 6 models to produce improved results. These models generally produce more accurate predictions than a single model. From Table 4 and Table 5, we see that that the Ensemble Model with interaction terms has significantly better results compared to the Ensemble Model without interaction terms. Therefore, based on the validation evaluation metrics from Table 4 and 5, we chose the Ensemble Model with the interaction terms as our best model.

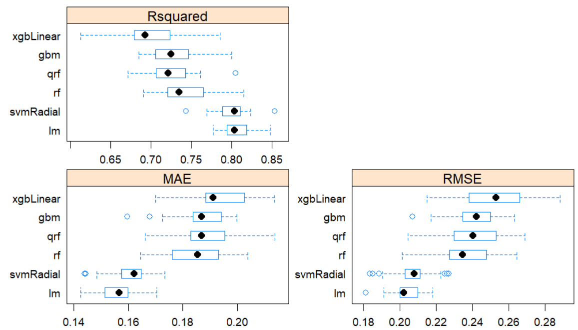

Fig. 15: Comparing In-sample Prediction Performance for Different Models Without Interaction Terms

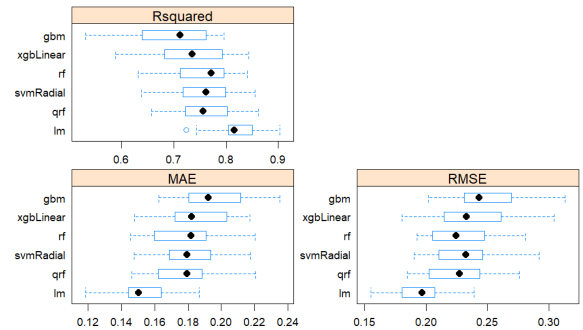

Fig. 16: Comparing In-sample Prediction Performance for Different Models With Interaction Terms

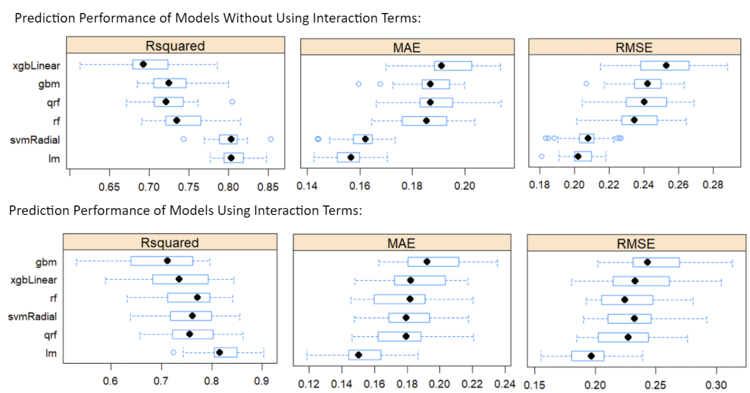

Fig. 17: Comparing Prediction Performance for Different Models Without and With Interaction Terms

5. Model Results

From Table 4, we see that the ensemble model have the lowest RMSE , meaning it has the best predictive performance out of all the models with no interaction terms.

Table 4: Comparing Prediction Performance of Different Models Without the Interaction Terms

Models	**RMSE**
Linear Model	0.203
Quantile Regression Forest	0.256
Random Forest	0.255

Models	RMSE
XGBoost Linear	0.255
Radial Support Vector Machine	0.218
Gradient Boosting Machine	0.262
Ensemble Model	**0.1986**

From Table 5, we see that once again that the ensemble of all the 6 models listed has the lowest RMSE, meaning it has the best predictive performance out of all the individual models with interaction terms.

Table 5: Comparing Prediction Performance of Different Models With the Interaction Terms

Models	RMSE
Linear Model	0.196
Quantile Regression Forest	0.227
Random Forest	0.234
XGBoost Linear	0.242
Radial Support Vector Machine	0.267
Gradient Boosting Machine	0.274
Ensemble Model	**0.1878**

From Table 6, we use our best performed models on test dataset to evaluate whether our models are still valid when applying on test dataset, we can see that the RMSE from both models are quqite consistent with their result in Training dataset in Table 4 and 5.

Table 6: Prediction Performance of Best Models on Testing Dataset

Models	RMSE
Ensemble Model Without Interaction Terms	0.192664
Ensemble Model With Interaction Terms	0.2042341

The two plots below show how much the prediction from our best 2 models deviated from actual value therefore giving us a rough estimation of whether a model is a good fit or not. We clearly see that both models'prediction are quite near the actual values, most points were near to the fitted line indicating a good fit.

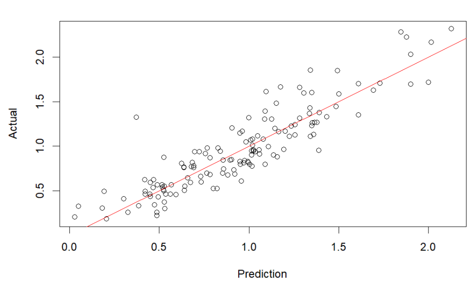

Fig. 18: Predicted vs Actual Values (Ensemble Model with interaction terms)

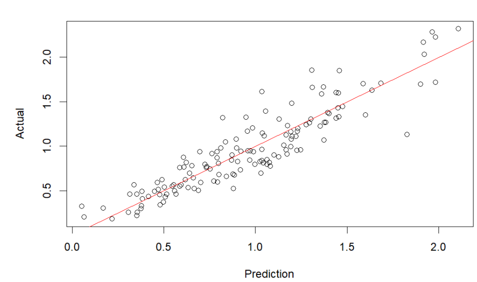

Fig. 19: Predicted vs Actual Values (Ensemble Model without interaction terms)

Comparing the predictive performance of the best models from both Table 4 and Table 5, we see that the latter (Ensemble Model, RMSE: 0.1795) has a better performance. Therefore, the best model we chose for making predictions was the Ensemble Model with the interaction terms.

6. Outlier Detection

We examined the stream flow values in our dataset by using the interquartile range (IQR) method, classifying any stream flow values more than 1.5 times the IQR below the first quartile or above the third quartile as an outlier. Looking at Fig. 20, there are 16 extreme high points that are outliers but no extreme low points. Grouping stream flow values by gridcodes is looking at each watershed separately. The average stream flow values vary from watershed to watershed. The average stream flow values for watershed at grid code 14 is higher than the rest of the watersheds. On the other hand, the average stream flow values for watershed at grid code 1264 is lower than the rest of the locations. Gridcodes 1212 and 1391 have the most outliers and it is worth looking into and learning more about those areas.

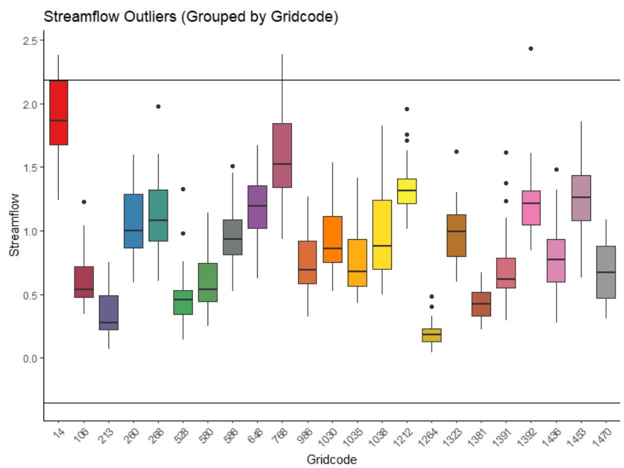

Fig. 20: Streamflow Outliers vs Gridcode

7. Limitations

- After feature selection, there is some potential that the best features for each model type were not selected. When selecting the features that would be included in the hyper-parameter tuning for the models we used the results of both Boruta (random forest based selection method) and Forward Stepwise Regression (regression based selection method). The selected features from each both agreed with each other so they seemed reliable, however we did not do an exhaustive search over all variables for each of the models and it is possible there were better combinations.
- Gridcode being selected as an important feature may lead to poor model performance due to lack of data to properly fit especially in the 'with interaction' case. There are 23 separate gridcodes, so there are only 20-40 observations for each gridcode which is not very much (especially for tree based models). Having access to more data could result in much higher performance.
- Each observation in the dataset is an aggregation of climate data collected over the year (i.e. they are the average values collected over the year). This limits the forecasting power of the predictive models that we have fit as we would need to use the forecasted explanatory variables to predict the stream flow, which will most likely lower the performance of the model.

8. Conclusion

After all the model iterations and improvements, we were able to achieve fairly good results with the Ensemble Model taking into account the interaction effect between variables. These results have large implications when it comes to water resource management economically. We have conducted our primary analysis taking into account the spatial data (e.g. gridcodes) which serves as a good MVP to predict the streamflow.

As a side interest and to expand upon the idea of predicting the streamflow solely using climate variables and not any spatial and temporal data, we conducted an analysis and trained models without this data and the predictive performance dropped significantly. Although this addresses the client's first research question of whether one catchment can be used to extrapolate stream flow in another catchment, the limitation we faced was lack of training data. To be able to model such a complex problem using climatic variables, we need more data to train our model which can help with improving the predictive performance of the model.

To address the second research question of whether or not we can detect the unusual streamflow activity accurately, we built a proof of concept pipeline for the outlier detection system. It will take the features from our prediction model as input and label the observations as either anomalous or regular depending on the anomaly scores which are the measures of deviation from normal behavior. We will face the same challenge here - lack of training data. This will lead to an increase in false positives and false negatives in the outlier detection system which can have detrimental consequences. For example, not being able to detect a subtle increase in the streamflow (false negatives) which could lead to irrigation problems and in severe cases even floods. Or getting a huge pool of outlier values (false positives) that will raise false alarms of anomalous behavior more often than desired.

9. Future Research

There are two areas of further research that would help address the limitations in this study; first is further investigating outliers and anomaly detection and second is training a model on a more granular time frame.

- The outlier detection that we have done is appropriate for identifying outliers in the current dataset but does not have a real-world application. Training a classification model to predict extreme values in streamflow and/or training an anomaly detection model to find unusual patterns in streamflow and the climate variables. Both models could have more real-world application in flood/drought prevention which is useful in fields such as agriculture.
- Additionally, requiring input data for the whole year for the model makes it impractical to accurately predict streamflow values in the future as we would need to use forecasted values for the input variables. Having the data be on a more granular time frame would greatly benefit the analysis and real-world predictive power of the model as we could train the predictive streamflow model only using past variables. The resulting model would not rely on using forecasted input variables, which addresses a critical limitation of this study.

10. References

- Government of Canada / Gouvernement du Canada. (2021, November 25). Government of Canada / gouvernement du Canada. Climate. Retrieved February 5, 2022, from https://climate.weather.gc.ca/glossary_e.html
- US Department of Commerce, N. O. A. A. (2012, March 8). Snow measurement guidelines. Snow Measurement Guidelines. Retrieved February 5, 2022, from https://www.weather.gov/gsp/snow
- Janssen, J., & Ameli, A. A. (2021). A Hydrologic Functional Approach for Improving Large-Sample Hydrology Performance in Poorly Gauged Regions. Water Resources Research, 57(9), e2021WR030263.
- Statistical interaction: More than the sum of its parts. Statistics Solutions. (2021, June 22). Retrieved February 21, 2022, from https://www.statisticssolutions.com/statistical-interaction-more-than- the-sum-of-its-parts/

11. Appendix I (For Client)

A.1.1 Interaction Term

> "In statistics, an interaction is a special property of three or more variables, where two or more variables interact to affect a third variable in a non-additive manner. In other words, the two variables interact to have an effect that is more than the sum of their parts."

Reference: Statistical interaction: More than the sum of its parts. Statistics Solutions. (2021, June 22). Retrieved February 21, 2022, from https://www.statisticssolutions.com/statistical-interaction-more-than-the-sum-of-its-parts/

A.1.2. Important Variables using Forward Stepwise Regression (including interaction terms)

gridcode	PREC
SDEN	EVA
gridcode:SFAL	TEMP
PEVA	SFAL
TEMP:PREC	SDEN:SFAL
SFAL:TEMP	SDEN:PREC
EVA:TEMP	gridcode:PEVA
PEVA:MTEMP	gridcode:PREC
gridcode:EVA	EVA:SDEN
gridcode:SDEN	SDEN:TEMP

12. Appendix II (For Mentor)

12.1. Installing Packages

```
#Sys.setenv(LANG = "en")
# library(tidyverse)
# library(ggplot2)
# library(GGally)
# library(DataExplorer)
# library(olsrr)
# library(gridExtra)
# library(cluster)
# library(factoextra)
# library(caretEnsemble)
# library(caret)
# library(mlbench)
# library(Metrics)
# library(Boruta)
# library(tidymodels)   # packages for modeling and statistical analysis
# library(tune)         # For hyperparemeter tuning
# library(workflows)    # streamline process
# library(tictoc)
# library(quantregForest)
# library(e1071)
# library(solitude)
# library(RColorBrewer)
```

12.2. Loading the Data

```
# Loading the data
# dt = read_csv('data_stat450.csv')
#
# dt2 = dt

# # Removing 'year' column
# dt = dt[-c(2)]
#
# # Renaming column names for simplicity
# colnames(dt) = c('gridcode','EVA','PEVA', 'SDEN','SDEP','SWEQ','SFAL','SMELT','TEMP',
#                  'PREC','Q')
# colnames(dt2) = c('gridcode','year','EVA','PEVA', 'SDEN','SDEP','SWEQ','SFAL','SMELT'
#                       ,'TEMP', 'PREC','Q') # for anomaly detection
#
# # Converting gridcode to a factor
# dt$gridcode = as.factor(dt$gridcode)
#
# head(dt)
# length(unique(dt$gridcode)) # 23
```

12.3. Explanatory Data Analysis

```
# Generating summary statistics for dt
# summary(dt)
#
# # EDA
# plot_intro(dt)
# plot_missing(dt)
# plot_bar(dt)
# plot_histogram(dt)
# plot_density(dt)
# plot_qq(dt)
# plot_qq(dt, by = "gridcode")
# plot_correlation(dt)
# plot_boxplot(dt, by = "gridcode")
# plot_scatterplot(split_columns(dt)$continuous, by = "Q")
# plot_prcomp(na.omit(dt), maxcat = 4L)
#
# # Checking dimensions of dt
# nrow(dt) # 774
# ncol(dt) # 11
```

12.3.1. Examining Variables by Gridcode

```
# look at Stream Flow vs Mean Potential Evaporation
# set.seed(123)
# dt.valid <- na.omit(dt)
# gridcode_sample <- dt.valid |> select(gridcode) |> distinct()
# gridcode_sample <- sample(gridcode_sample$gridcode, 9)
# dt.valid |>
#     filter(gridcode %in% gridcode_sample) |>
```

```r
#     ggplot(aes_string(x="PEVA", y="Q")) +
#     geom_point() +
#     facet_wrap(~gridcode) +
#     geom_smooth(method="lm", se=FALSE, formula=y~x) +
#     labs(title="Stream Flow vs Mean Potential Evaporation") +
#   ylab('Stream Flow') +
#   xlab('Mean Potential Evaporation') +
#   theme(plot.caption = element_text(hjust = 0))
```

12.4. Data Preprocessing

```r
# labels <- paste(colnames(dt[,2:11]))
# boxplot(dt[,2:11],  xaxt = "n",xlab = "",
#         main = 'Comparing different explanatory variables (before scaling)')
# axis(1, labels = FALSE)
# text(x =  seq_along(labels), y = par("usr")[3] - 1, srt = 60, adj = 1,
#      labels = labels, xpd = TRUE)
```

```r
# Transformations

## right-skewed
#dt$Q = sqrt(dt$Q)
# dt$EVA = (dt$EVA)^(1/3)
# dt$SDEN = sqrt(dt$SDEN)
# dt$SDEP = log(dt$SDEP)
# dt$SWEQ = log(dt$SWEQ)
# dt$SFAL = sqrt(dt$SFAL)
# dt$SMELT = sqrt(dt$SMELT)
#
# # left-skewed
# dt$PEVA = sqrt(2000-dt$PEVA)
#
# plot_density(dt|> select(EVA,SDEN,SDEP, SWEQ, SFAL, SMELT,PEVA))
```

12.4.1. Transformations

```r
# set.seed(2020)
# rec <- recipe(Q ~.,
#               data = dt[,2:11]) %>%
#   step_corr(all_predictors()) %>%                 # removing highly correlated features
#   # Z-Score Standardization
#   step_center(all_numeric(), -all_outcomes())%>%   # centering data at mean = 0
#   step_scale(all_numeric(), -all_outcomes())       # scaling data with variance  = 1
#
# trained_rec =  prep(rec, training = dt, retain = TRUE)
# dt_prep = cbind(gridcode = dt$gridcode, as.data.frame(juice(trained_rec)))
#
# # Separating the actual test set w/o labels
# main_dt <- na.omit(dt_prep)
# test_nolabel_df <- dt_prep[is.na(dt$Q),]
#
```

```
# main_dt
#
# labels <- paste(colnames(main_dt[,2:8]))
# boxplot(main_dt[,2:8],  xaxt = "n",xlab = "")
# axis(1, labels = FALSE)
# text(x =  seq_along(labels), y = par("usr")[3] - 1, srt = 60, adj = 1,
#      labels = labels, xpd = TRUE)
```

12.4.2. Preprocessing Pipeline

```
# plot_scatterplot(split_columns(main_dt)$continuous, by = "Q")
#
# ggpairs(main_dt[,2:8],lower = list(continuous = wrap("smooth", alpha = 0.3,
#                                                     size=0.05)),
#        upper = list(continuous = wrap("cor", size=2)))
```

12.4.3. Relationship b/w explanatory variables & the response

```
# boruta_output <- Boruta(Q ~ ., data=na.omit(main_dt), doTrace=0)
# roughFixMod <- TentativeRoughFix(boruta_output)
# boruta_signif <- getSelectedAttributes(roughFixMod)
#
# # Variable Importance Scores
# imps <- attStats(roughFixMod)
# imps2 = imps[imps$decision != 'Rejected', c('meanImp', 'decision')]
# head(imps2[order(-imps2$meanImp), ])  # descending sort
#
# # Plot variable importance
# plot(boruta_output, cex.axis=.7, las=2, xlab="", ylab = "Variable Importance")
#
# selected_features <- c('PREC','gridcode','EVA','TEMP','PEVA','SFAL','SDEN')
#
# main_fs <- main_dt[,(colnames(main_dt) %in% append(selected_features, "Q"))]
# main_fs
```

12.4.4. Feature Selection Using boruta

```
# dt_interaction <- main_dt
# FS.lm <- lm(Q ~ (.)^2, data = dt_interaction)
#
# OLS <- ols_step_forward_p(FS.lm)
# OLS
# plot(OLS)
```

12.4.5. Feature Selection Using Forward Stepwise Regression

12.5. Model Training

```
# set.seed(123)
# split <- createDataPartition(y=main_fs$Q, p=.8,list=F)
# train <- main_fs[split,]
```

```
# nrow(train)
# test <- main_fs[-split,]
# nrow(test)
```

12.5.1. Splitting Data into Train/Test sets

```
# set.seed(123)
# my_control = trainControl(method = 'repeatedcv', # for "cross-validation",
#                              repeats = 3,
#                              number = 10, # number of k-folds
#                              savePredictions = 'final',
#                              search = 'random')
#
# model_list1 = caretList(Q~.,
#                           data = train,
#                           methodList = c('lm', 'rf', 'qrf', 'xgbLinear','svmRadial',
#                                          'gbm'),
#                           tuneList = NULL)
#
# ensemble1 = caretEnsemble(model_list1,
#                             metric = 'RMSE',
#                             trControl = my_control)
```

12.5.2. Cross-Validation (w/o Interaction terms)

```
# model_list2 = caretList(Q ~ gridcode+PREC+SDEN+EVA+gridcode:SFAL+TEMP+PEVA+SFAL+
#                            TEMP:PREC+SDEN:SFAL+
#                            SFAL:TEMP+SDEN:PREC+EVA:TEMP+gridcode:PEVA+
#                            PEVA:TEMP+gridcode:PREC+gridcode:EVA+
#                            EVA:SDEN+gridcode:SDEN+SDEN:TEMP,
#                   data = train,
#                           trControl = my_control,
#                           methodList = c('lm', 'rf', 'qrf', 'xgbLinear',
#                           'svmRadial', 'gbm'),
#                           tuneList = NULL)
#
# ensemble2 = caretEnsemble(model_list2,
#                             metric = 'RMSE',
#                             trControl = my_control)
```

12.5.3 Cross-Validation (w/ Interaction terms)

```
# options(digits = 3)
# model_results1 = data.frame(
#   LM = mean(model_list1$lm$results$RMSE),
#   QRF = mean(model_list1$qrf$results$RMSE),
#   RF = mean(model_list1$rf$results$RMSE),
#   XGBL = mean(model_list1$xgbLinear$results$RMSE),
#   SVMR = mean(model_list1$svmRadial$results$RMSE),
#   GBM = mean(model_list1$gbm$results$RMSE)
# )
```

```
#
# best_model_train1 = apply(model_results1, 1, FUN = mean)
# print(model_results1)
#
# resamples1 <- resamples(model_list1)
# resamples1
# summary(resamples1)
# dotplot(resamples1, metric = 'RMSE')
# modelCor(resamples1)
#
# # Ensemble Model Results
# summary(ensemble1)
# plot(ensemble1)
#
# scales1 = list(x=list(relation='free'), y=list(relation='free'))
# bwplot(resamples1,scales = scales1,layout = c(2,2))
```

12.5.4 Evaluation Metrics (w/o Interaction terms)

```
# options(digits = 3)
# model_results2 = data.frame(
#   LM = mean(model_list2$lm$results$RMSE),
#   QRF = mean(model_list2$qrf$results$RMSE),
#   RF = mean(model_list2$rf$results$RMSE),
#   XGBL = mean(model_list2$xgbLinear$results$RMSE),
#   SVMR = mean(model_list2$svmRadial$results$RMSE),
#   GBM = mean(model_list2$gbm$results$RMSE)
# )
#
# best_model_train2 = apply(model_results2, 1, FUN = mean)
# print(model_results2)
#
# resamples2 <- resamples(model_list2)
# resamples2
# summary(resamples2)
# dotplot(resamples2, metric = 'RMSE')
# modelCor(resamples2)
#
# # Ensemble Model Results
# summary(ensemble2)
# plot(ensemble2)
#
# scales2 = list(x=list(relation='free'), y=list(relation='free'))
# bwplot(resamples2,scales = scales2,layout = c(2,2))
```

12.5.5 Evaluation Metrics (w/ Interaction terms)

12.6 Predictions

```
# # PREDICTIONS
# pred_lm1 <- predict.train(model_list1$lm, newdata = test)
# pred_qrf1 <- predict.train(model_list1$qrf, newdata = test)
```

```
# pred_rf1 <- predict.train(model_list1$rf, newdata = test)
# pred_xgbL1 <- predict.train(model_list1$xgbLinear, newdata = test)
# pred_svmr1 <- predict.train(model_list1$svmRadial, newdata = test)
# pred_gbm1 <- predict.train(model_list1$gbm, newdata = test)
# predict_ens1 <- predict(ensemble1, newdata = test)
#
# # RMSE
# y_test = test[,8]
# pred_RMSE1 <- data.frame(ENS = RMSE(predict_ens1, y_test),
#                          LM = RMSE(pred_lm1, y_test),
#                          QRF = RMSE(pred_qrf1, y_test),
#                          RF = RMSE(pred_rf1, y_test),
#                          XGBL = RMSE(pred_xgbL1, y_test),
#                          SVMR = RMSE(pred_svmr1, y_test),
#                          GBM = RMSE(pred_svmr1, y_test))
#
# print(pred_RMSE1)
#
# best_model_test1 = apply(pred_RMSE1, 1, FUN = mean)
#
# pred_cor1 <- data.frame(ENS = cor(predict_ens1, y_test),
#                         LM = cor(pred_lm1, y_test),
#                         QRF = cor(pred_qrf1, y_test),
#                         RF = cor(pred_rf1, y_test),
#                         XGBL = cor(pred_xgbL1, y_test),
#                         SVMR = cor(pred_svmr1, y_test),
#                         GBM = cor(pred_gbm1, y_test),
#                         ENS = cor(predict_ens1, y_test))
#
# print(pred_cor1)
# ```
# ```{r}
# # par(mfrow = c(3,3))
# # plot(pred_lm1, y_test) + abline(0,1, col = 'red')
# # plot(pred_qrf1, y_test) + abline(0,1, col = 'red')
# # plot(pred_rf1, y_test) + abline(0,1, col = 'red')
# # plot(pred_xgbL1, y_test) + abline(0,1, col = 'red')
# # plot(pred_svmr1, y_test) + abline(0,1, col = 'red')
# # plot(pred_gbm1, y_test) + abline(0,1, col = 'red')
# plot(predict_ens1, y_test, xlab="Prediction",ylab="Actual") +
#     abline(0,1, col = 'red')
```

12.6.1. Predictions (w/o Interaction terms)

```
# # PREDICTIONS
# pred_lm2 <- predict.train(model_list2$lm, newdata = test)
# pred_qrf2 <- predict.train(model_list2$qrf, newdata = test)
# pred_rf2 <- predict.train(model_list2$rf, newdata = test)
# pred_xgbL2 <- predict.train(model_list2$xgbLinear, newdata = test)
# pred_svmr2 <- predict.train(model_list2$svmRadial, newdata = test)
# pred_gbm2 <- predict.train(model_list2$gbm, newdata = test)
# predict_ens2 <- predict(ensemble2, newdata = test)
#
```

```
# # RMSE
# y_test = test[,8]
# pred_RMSE2 <- data.frame(ENS = RMSE(predict_ens2, y_test),
#                          LM = RMSE(pred_lm2, y_test),
#                          QRF = RMSE(pred_qrf2, y_test),
#                          RF = RMSE(pred_rf2, y_test),
#                          XGBL = RMSE(pred_xgbL2, y_test),
#                          SVMR = RMSE(pred_svmr2, y_test),
#                          GBM = RMSE(pred_gbm2, y_test))
#
# print(pred_RMSE2)
#
# best_model_test2 = apply(pred_RMSE2, 1, FUN = mean)
#
# pred_cor2 <- data.frame(ENS = cor(predict_ens2, y_test),
#                         LM = cor(pred_lm2, y_test),
#                         QRF = cor(pred_qrf2, y_test),
#                         RF = cor(pred_rf2, y_test),
#                         XGBL = cor(pred_xgbL2, y_test),
#                         SVMR = cor(pred_svmr2, y_test),
#                         GBM = cor(pred_gbm2, y_test),
#                         ENS = cor(predict_ens2, y_test))
#
# print(pred_cor2)
# ```
# ```{r}
# # par(mfrow = c(3,3))
# # plot(pred_lm2, y_test) + abline(0,1, col = 'red')
# # plot(pred_qrf2, y_test) + abline(0,1, col = 'red')
# # plot(pred_rf2, y_test) + abline(0,1, col = 'red')
# # plot(pred_xgbL2, y_test) + abline(0,1, col = 'red')
# # plot(pred_svmr2, y_test) + abline(0,1, col = 'red')
# # plot(pred_gbm2, y_test) + abline(0,1, col = 'red')
# plot(predict_ens2, y_test, xlab="Prediction",ylab="Actual")
#   + abline(0,1, col = 'red')
```

12.6.2. Predictions (w/ Interaction terms)

12.7. Anomaly Detection

```
# train_2 = filter(train, gridcode == 14 | 768)
# q = train_2 |> select(Q) |> data.frame()
# train_2 = select(train_2, -Q)
#
# iforest = isolationForest$new()
# train_2 = na.omit(train_2)
# train_2$gridcode = as.factor(train_2$gridcode)
# train_2[,2:7] = data.frame(scale(train_2[,2:7], TRUE, TRUE))
#
# iforest$fit(train_2)
# train_2$pred = iforest$predict(train_2)
# train_2$label = as.factor(ifelse(train_2$pred$anomaly_score >=0.64,
#                                     "anomaly", "normal"))
#
```

```r
# barplot(table(train_2$label),
# xlab = "Class",
# col = c("red","blue")
# )
#
# qplot(train_2$pred$anomaly_score, q$Q, color = train_2$label)
#
# filter(train_2, label=="anomaly")
```

12.8. Outlier Detection

```r
# getPalette = colorRampPalette(brewer.pal(9, "Set1"))
#
# dt.valid <- na.omit(dt2)
# qt <- quantile(dt.valid$Q, na.rm = TRUE)
# iqr <- qt[4]-qt[2]
# upper.bd <- qt[4]+iqr*1.5
# lower.bd <- qt[2]-iqr*1.5
#
# ggplot(dt.valid, aes(y=Q, group=year,x=year)) +
#   geom_boxplot() +
#   geom_hline(yintercept = upper.bd) +
#   geom_hline(yintercept = lower.bd) +
#   ylab("Streamflow") +
#   ggtitle("Streamflow Outliers (Grouped by Year)")
```