Curso R Redbioma

EVALUACIÓN DE TRES MODELOS PARA EL AJUSTE DE DATOS Y PREDICCIÓN DE CAUDALES EN FUNCIÓN DE LA PRECIPITACIÓN PARA LA CUENCA DEL RÍO TEMPISQUE.

Proyecto Final

Leonel Sanabria Méndez

2024

Objetivos

Principal: Desarrollar y evaluar tres modelos de ajuste y predicción de caudal en función de la precipitación en la cuenca del río Tempisque, Costa Rica, comparando su eficiencia frente a datos observados y simulados mediante modelos hidrológicos conceptuales.

Específicos:

1- Diseñar tres modelos de predicción (lineal, exponencial y de árbol de decisión tipo Gradient Boosting) para estimar el caudal a partir de datos de precipitación en la cuenca del río Tempisque, utilizando datos del período 1977-2020 en R.
2- Evaluar y comparar la eficiencia de los modelos de predicción (lineal, exponencial y Gradient Boosting) frente a datos observados, analizando el desempeño de los modelos lineal y exponencial en R.
3- Seleccionar el modelo con mejor desempeño y utilizarlo para predecir caudales en el período 2021-2099 a partir de datos de precipitación generados por un modelo de circulación general con reducción de escala, utilizando R para comparar estos resultados con caudales simulados por el modelo hidrológico HBV-Light.

Método

Para el desarrollo de los modelos se utilizan datos de caudal observados por la estación Guardia 19-01, ubicada en la cuenca media del río Tempisque en Liberia, Guanacaste, Costa Rica, para el periodo de 1977-2020. Además, se emplean datos de precipitación media obtenidos de las estaciones meteorológicas disponibles en la cuenca de aporte.

Estos datos se disponen en una matriz de Excel clasificada por día, mes y año, los cuales se cargan en R mediante un tibble, siguiendo el siguiente procedimiento:

1. Carga de los paquetes a utilizar: readxl, tidyverse, tidymodels, readr, skimr, xgboost y dplyr.
2. Creación del tibble.
3. Generación de un modelo descriptivo y su significancia como preámbulo para la creación de modelos predictivos.
4. Ploteo de los datos y generación de un modelo lineal gráficamente como preámbulo.
5. Generación de modelos predictivos, asignando datos de entrenamiento y prueba.
6. Generación de modelos: lineal, exponencial y de árbol de decisión tipo Gradient Boosting, aplicando la siguiente ruta: receta, flujo de trabajo, ajuste y evaluación del modelo mediante el coeficiente de determinación R^2 y el RMSE.

Siguiendo el mismo procedimiento, se efectúa una comparación entre los modelos lineal y exponencial.

Finalmente, se dispone de una matriz con datos de precipitación obtenidos de un modelo de circulación general con reducción de escala para la región de la cuenca media del río Tempisque, para el periodo de 2021 a 2099. Estos datos permiten realizar una modelación con el modelo hidrológico HBV-Light, obteniendo predicciones de caudal para el periodo 2021-2099.

Se generan dos nuevas columnas, una que muestra las predicciones para caudales obtenidos a partir del mejor modelo obtenido con R y otra que muestra el error absoluto; además, se evalúa la relación entre datos de ambos modelos de predicción mediante el coeficiente de determinación R^2.

Resultados

Cuadro 1: Ecuación y criterios de evaluación obtenidos para cada modelo

Modelo	Ecuación	R^2	RMSE
Lineal	Q= 1.6672P+0.1448	0.253	0.311
Exponencial	$Q=1.2790e^{0.2009P}$	0.141	3.97
Gradient Boosting	$Q = \sum_{m=1}^{500} \gamma_m h_m P$	0.200	3.83

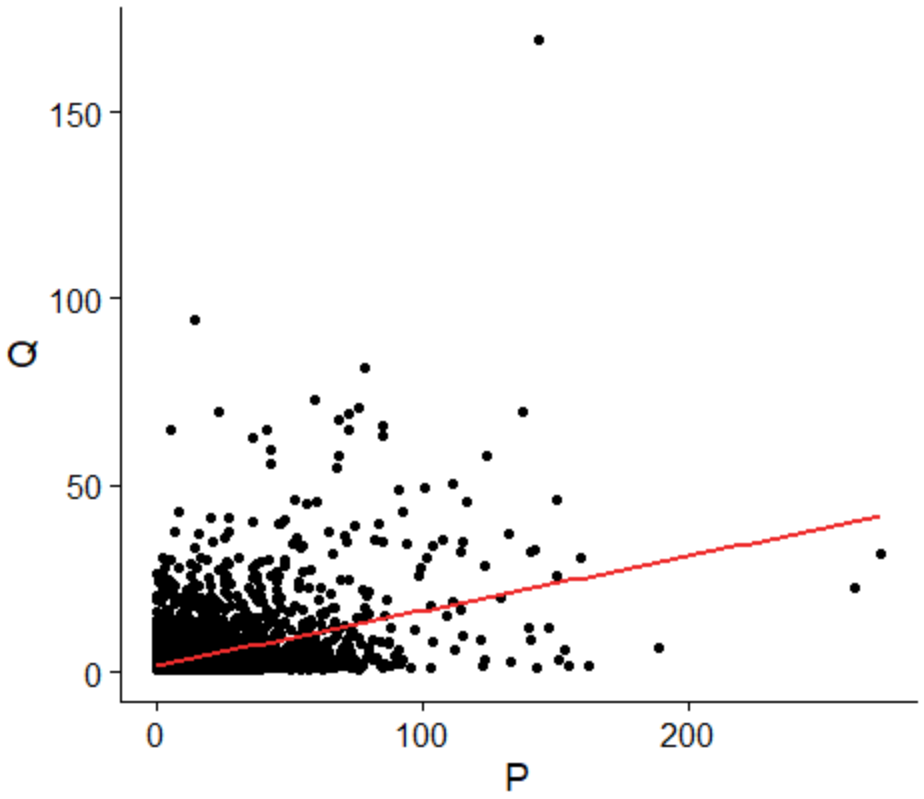

Figura 1: Ploteo de datos y generación de ajuste lineal

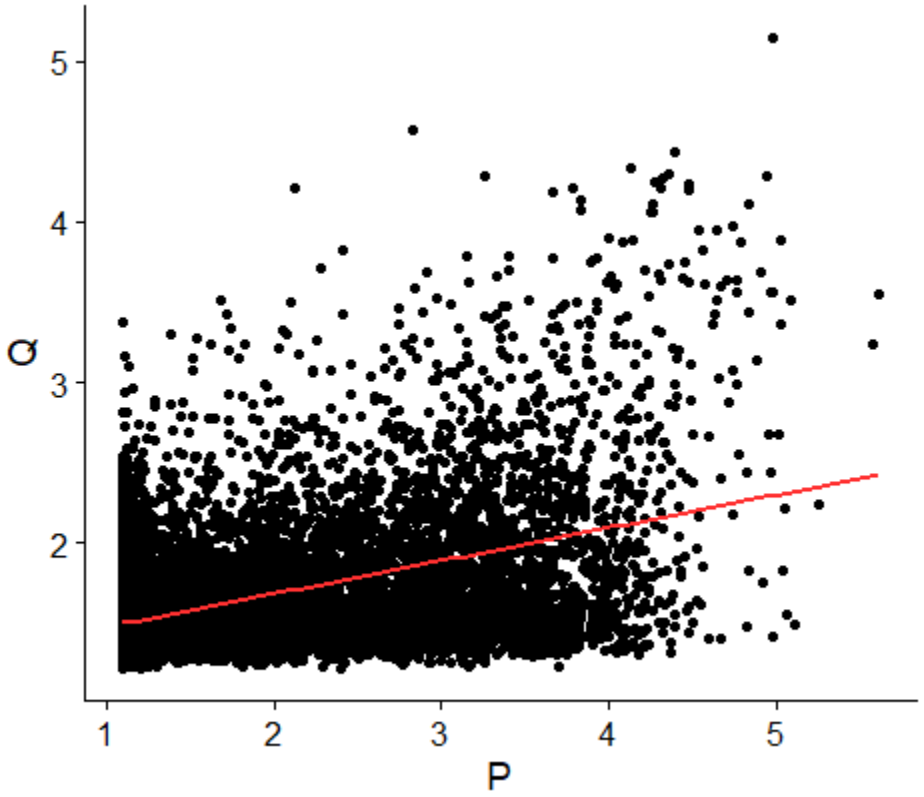

Figura 2: Ploteo de datos a escala logarítmica.

Cuadro 2: Comparación entre modelos

Modelos	Ecuación	R^2	RMSE
Mejor modelo lineal	$Q= 1.6672P+0.1448$	0.253	0.311
Mejor modelo exponencial	$Q=1.2790e^{0.2009P}$	0.141	3.97

Cuadro 3: Evaluación entre mejor modelo lineal y modelo hidrológico conceptual.

Criterio de comparación	R^2
Modelo lineal vs Modelo hidrológico (HBV-Light)	0.2471379

Discusión

Los tres modelos evaluados presentan una baja eficiencia en la predicción de caudales a partir de datos de precipitación. Sin embargo, el modelo que mejor se ajusta al patrón observado en estos datos es el modelo lineal. Desde el preámbulo, al momento de graficar los datos, se observa una alta dispersión, lo que complica el ajuste del modelo.

La significancia de los datos simulados con el modelo lineal, al compararlos con los datos observados, es muy similar a la significancia de los datos simulados mediante el modelo lineal, en comparación con los datos generados por el software HBV-Light para la predicción de caudal en un escenario futuro. Esta situación es esperable, ya que los datos generados por HBV-Light presentan una significancia, medida a través del R^2, con los datos observados de aproximadamente 0.80.

Este ensayo demuestra que los modelos lineales, exponenciales y de árboles de decisión no son suficientes para ajustar el comportamiento del cauce del río Tempisque en función de la precipitación. Es necesario explorar otros enfoques, especialmente modelos multivariables que incluyan otros factores como temperatura y evapotranspiración. Otra alternativa es la calibración de modelos conceptuales que involucren múltiples parámetros, como HBV-Light, el cual evalúa la cobertura del suelo, precipitación, temperatura, evapotranspiración, y flujos de agua superficial y subsuperficial, entre otros.

Anexos

```
#Carga de paquetes

library(readxl)

library(tidyverse)

library(tidymodels)

library(readr)

library(skimr)

install.packages("xgboost")

library(xgboost)

library(dplyr)

#Carga de datos

datos <- read_excel("datos_obs_1977_2020.xlsx")


#Modelo descriptivo(lineal)

modQP <- linear_reg() |>

  fit(Q ~ P, data = datos)

modQP

#significancia

tidy(modQP)

#ploteo de datos, lineal

datos |>

  ggplot(aes(x = P,

             y = Q)) +
```

```
  geom_point() +
  geom_smooth(method = lm,
              se = FALSE,
              col = "firebrick2") +
  cowplot::theme_cowplot()
#Modelo predictivo
#semilla
set.seed(5)
#asigno datos de calentamiento
split_datos<- initial_split(datos, prop = 0.75)
# datos de entrenamiento y prueba:
train_data <- training(split_datos)
test_data  <- testing(split_datos)
#receta
receta <- recipe(Q ~ P,
                 data = train_data)
#asiganar el modelo
mod_rl <- linear_reg() |>
  set_engine("glm")
#Flujo de trabajo
QP_wflow <-
  workflow() |>
  add_model(mod_rl) |>
```

```
  add_recipe(receta)
#ajuste de modelo
QP_fit <-
  QP_wflow |>
  fit(data = train_data)
#predecir
predict(QP_fit, test_data)
#comparar con datos observados
test_preds <- QP_fit |>
  augment(test_data)
test_preds
#Evaluar modelo
eval_metrics <- metric_set(rmse,rsq_trad)


test_preds |>
  eval_metrics(truth = Q,
               estimate = .pred)
#ploteo de datos, exponencial
datos |>
  mutate(across(c(P, Q), ~log(.x+3))) |>
  ggplot(aes(x = P,
             y = Q)) +
  geom_point() +
```

```
  geom_smooth(method = lm,
              col = "firebrick1",
              se = FALSE) +
  cowplot::theme_cowplot()
#Modelo exponencial
#Receta
receta2 <- recipe(Q ~ P,
                  data = train_data) |>
  step_mutate_at(all_numeric(),
                 fn = ~log(.x+3))
#asignar modelo
show_engines("linear_reg")

mod_rl2 <- linear_reg() |>
  set_engine("glm")

#Flujo de trabajo
QP_wflow2 <- workflow() |>
  add_model(mod_rl2) |>
  add_recipe(receta2)

#Ajuste de modelo
QP_fit2 <-
```

```
  QP_wflow2 |>
  fit(data = train_data)
#Predecir
test_preds2 <- QP_fit2 |>
  augment(test_data)
test_preds2
#Métricas de evaluación
eval_metrics <- metric_set(rmse,rsq_trad)

test_preds2 |>
  mutate(across(.pred, ~exp(.x)-3)) |>
  eval_metrics(truth = Q,
               estimate = .pred)
#Modelo arbol de decisión, gradient boosting
#Receta
receta_gb <- recipe(Q ~ P, data = train_data) |>
  step_impute_mean(all_numeric_predictors()) |>
  step_normalize(all_numeric_predictors())

#asignar modelo
show_engines("boost_tree")

mod_gb <- boost_tree(
```

```
  trees = 500,
  learn_rate = 0.01,
  tree_depth = 6
) |>
  set_engine("xgboost") |>
  set_mode("regression")

#Flujo de trabajo
QP_wflow_gb <- workflow() |>
  add_model(mod_gb) |>
  add_recipe(receta_gb)

#Ajuste de modelo
QP_fit_gb <-
  QP_wflow_gb |>
  fit(data = train_data)
#Predecir
test_preds_gb <- QP_fit_gb |>
  augment(test_data)
test_preds_gb
#Métricas de evaluación
eval_metrics <- metric_set(rmse, rsq_trad)
```

```
test_preds_gb |>
  eval_metrics(truth = Q,
               estimate = .pred)


#comparación de modelos (lineal y exponencial)
set.seed(5)
dts <- datos |>
  dplyr::select(Q, P)


dts_split <- initial_split(dts,
                           prop = 0.75)
dts_train <- training(dts_split)
dts_test <- testing(dts_split)
dts_r <- vfold_cv(dts_train,
                  v = 10)


dts_r$splits


#Flujo de trabajo
resuls <- workflow_set(preproc = list(simple = receta,
                                      log = receta2),
                       models = list(lm = mod_rl)) |>
  workflow_map(fn = "fit_resamples",
```

```
            seed = 1101,
            verbose = TRUE,
            resamples = dts_r)
#desanidar resultados
resuls |>
  filter(wflow_id == "simple_lm") |>
  select(result) |>
  unnest(result)

#métricas de evaluación
resuls |>
  filter(wflow_id == "simple_lm") |>
  select(result) |>
  unnest(result) |>
  unnest(.metrics)
#seleccionar el mejor modelo lineal
mejor <- resuls |>
  extract_workflow_set_result("simple_lm") %>%
  select_best(metric = "rmse")

mod <- resuls |>
  extract_workflow("simple_lm") %>%
  finalize_workflow(mejor) %>%
```

```
  fit(data = dts_train)


mod |>
  augment(dts_train) |>
  eval_metrics(truth = Q,
               estimate = .pred)
#seleccionar el mejor modelo exponencial
mejor <- resuls |>
  extract_workflow_set_result("log_lm") %>%
  select_best(metric = "rmse")


mod <- resuls |>
  extract_workflow("log_lm") %>%
  finalize_workflow(mejor) %>%
  last_fit(split = dts_split)


test_preds <- mod  %>%
  collect_predictions()


test_preds |>
  mutate(across(c(Q, .pred), ~ exp(.x)-3)) |>
  eval_metrics(truth = Q,
               estimate = .pred)
```

```
#Carga de datos predichos a partir de MCG y HBV light

datos_sim <- read_excel("datos_sim_2021_2099.xlsx")


#Crear una nueva columna para los datos del modelo lineal

datos_sim <- datos_sim %>%

  mutate(Qsim_lin = 1.6672 * P + 0.1448)

datos_sim

#Crear una nueva columna para estimar error absoluto

datos_sim <- datos_sim %>%

  mutate(AE = Qsim_lin - Q)

datos_sim


#Estimación de Coeficiente de determinación (R²) entre datos del modelo lineal y HBV mediante modelo lineal

modelo <- lm(Q ~ Qsim_lin, data = datos_sim)

summary(modelo)$r.squared
```