

Camera based Water Stage and Discharge Prediction with Machine Learning

Ernesto Adrián Álvarez Salazar A00227490

Carlos Javier Leal Beltrán A01741355

Carlos Moisés Chávez Jiménez A01637322

Luis Armando Salazar López A01114901

Septiembre 2022 - Diciembre 2022

1 Introducción

La medición y el modelado precisos del nivel (la altura del nivel del agua en el río) y la descarga (el caudal del río) de la corriente son importantes para la gestión diaria del agua, el pronóstico y la gestión de inundaciones, la evaluación del cumplimiento de los acuerdos de uso del agua y el diseño de embalses, sistemas de suministro de agua, puentes y alcantarillas (Boiten, 2008). Los datos continuos de series de tiempo de medidores de flujo también son críticos para calibrar y/o validar modelos de aguas superficiales y subterráneas, mientras que las brechas en estos datos aumentan la incertidumbre en las predicciones de un modelo. El nivel de la corriente normalmente se mide con sensores flotantes, de presión, ópticos y acústicos (Turnipseed y Sauer, 2010). Estos sensores tradicionales pueden fallar y requerir un mantenimiento regular, los cuales son costosos.

Como resultado, es posible que se produzcan lagunas en los registros de flujo y descarga debido a una instalación incorrecta (p. ej., especialmente durante estudios a corto plazo, cuando las características del sitio no son bien conocidas), fallas en los equipos y/o brechas en el financiamiento de los programas de monitoreo. Por supuesto, las cámaras también pueden fallar, pero pueden proporcionar una redundancia económica con información que no está disponible en los sensores que emiten mediciones escalares individuales. Por ejemplo, las imágenes proporcionan una verificación visual de las condiciones hidrológicas, incluida la presencia de hielo, obstrucciones o cambios importantes en la geometría del canal. En este estudio, se opta por un enfoque al monitoreo pasivo que utiliza imágenes de lapso de tiempo que se pueden combinar con mediciones de sensores tradicionales que son adecuadas para llenar los vacíos en los registros de caudales.

2 Hipótesis

Utilizando la investigación previamente realizada por el equipo del socio formador, mediante optimización de las imágenes utilizables en la problemática implementando librerías de computer vision en python, a través del desarrollo de modelos matemáticos y el análisis de datos, se espera obtener formas distintas y optimizadas de abordar la problemática, modelos de regresión que nos permitan predecir el nivel y la descarga de una corriente con más eficacia y un modelo de clasificación que permita distinguir y optimizar entre imágenes utilizables dentro del proceso de obtención de datos, mejorando así en un aspecto general, la parte práctica en la resolución de esta problemática.

Ademas, comprobar que hay una diferencia detectable entre la predicción con imágenes de diferentes temporadas y en un lapso temporal pequeño (imágenes de entrenamiento y evaluación con no más de 1 año de diferencia) y una predicción con imágenes de la misma temporada pero con un lapso temporal largo (imágenes de entrenamiento y evaluación con más de 5 año de diferencia)

3 Tratamiento Inicial de los Datos

Para comenzar a trabajar con los datos, es necesario que pasen por un proceso de preparación que nos permita obtener la mejor parte de ellos. Este proceso se divide en tres etapas: Limpieza, Transformación y Visualización. A continuación desglosaremos las fases involucradas a este proceso:

3.1 Limpieza de los datos

La limpieza es la primera y una etapa fundamental del tratamiento de la información. Aquí se busca eliminar la mayor cantidad de imperfecciones que pudieramos llegar a encontrar. Cosas como valores faltantes, datos fuera de rango, dividir la información disponible en "entrenamiento" y "pruebas", eliminar columnas innecesarias para el análisis, etc.

SensorTime	CaptureTime	Filename	hlineAgency
SiteNumber	TimeZone	Stage	Discharge
CalcTimestamp	fNumber	isoSpeed	shutterSpeed
grayMean	graySigma	entropyMean	entropySigma
hMean	hSigma	sMean	sSigma
vMean	vSigma	areaFeatCount	grayMean 0
graySigma 0	entropyMean 0	entropySigma 0	hMean 0
hSigma 0	sMean 0	sSigma 0	vMean 0
vSigma 0	grayMean 0	graySigma 0	entropyMean 0
entropySigma 0	hMean 0	hSigma 0	sMean 0
sSigma 0	vMean 0	vSigma 0	WeirAngle
WeirPt1X	WeirPt1Y	WeirPt2X	WeirPt2Y
WwRawLineMin	WwRawLineMax	WwRawLineMean	WwRawLineSigma
WwCurveLineMin	WwCurveLineMax	WwCurveLineMean	WwCurveLineSigma

Variables disponibles en nuestro archivo de datos. (figura 3.1.1)

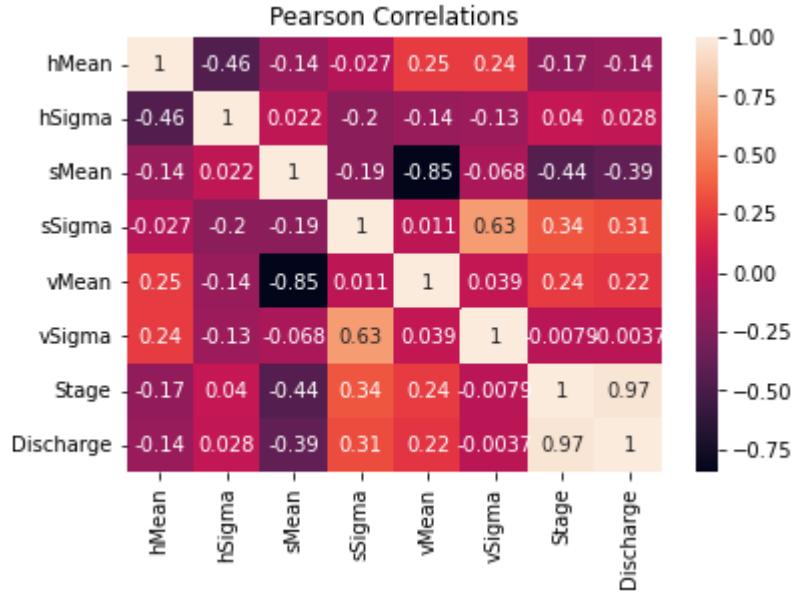
3.2 Transformación de los datos

Una vez revisados los datos, entendemos que debemos transformarlos para generar gráficos y diagramas que faciliten la búsqueda de patrones y la determinación de los parámetros necesarios para un modelo predictor de regresión y uno de clasificación.

Para esto, revisamos el conjunto de datos con la información proporcionada. Encontramos que primero debemos ordenar los datos con base en el timestamp. Estas variables que indican el tiempo con el que fueron realizadas las capturas de datos. Dentro del conjunto de datos se proporcionan tres variables para indicar el tiempo de captura: "CaptureTime", "SensorTime" y "CalcTimestamp", y cada una de estas contiene un formato distinto y a primera instancia nada fácil de trabajar, lo que indica que tenemos que separarlas para que queden enumeradas de una en una. Desarrollamos un método para tratar las fechas consiguiendo así, un orden en el formato de la captura de datos. En cuanto a lo demás, los datos venían en buenas condiciones, por lo que no tuvimos que arreglar valores nulos.

3.3 Visualización de los datos

Siendo las variables "Stage" y "Discharge" las que nos interesan predecir con un modelo, las comparamos con el resto de las variables para ir descartando las de menor correlación.



Ejemplo de las correlaciones de pearson realizadas. (figura 3.3.1)

Tras comparar con y analizar cada una de las variables dentro del conjunto de datos con las variables stage y discharge escogimos las siguientes variables, ya que cuentan con una correlación menor a -30 ó mayor a 30, por lo tanto tienen una correlación considerable para incluirse en un modelo: sSigma, grayMean 0, hMean 0, grayMean 1, hMean 1, sMean s, Sigma 1, WeirPt1Y, WeirPt2X, WeirPt2Y.

4 Generación y evaluación de los modelos de predicción para Discharge y Stage.

Tras identificar las variables que deseamos predecir y las variables que podemos utilizar para lograrlo, comenzamos a realizar modelos con el fin de obtener resultados similares a la investigación pasada, lo anterior con el fin de contar con un punto de partida.

Con la finalidad de mantener nuestros datos más limpios y prevenir que los modelos implementados caigan en un "overfitting" sepáramos nuestros datos en tres categorías: "Training" cuyo propósito es entrenar al modelo, "Testing" cuyo propósito es probar que tan bien entrenado se encuentra el modelo conforme a los datos del training y por último "Validation" cuyo propósito es presentar imágenes nuevas que validen el modelo una vez probado, con la finalidad de comprobar que el modelo reacciona bien ante datos nuevos.

4.1 Stage

Inicialmente generamos un modelo regresor con Random Forest sin limpiar los datos para comprobar la fiabilidad mostrada en la investigacion inicial. Los resultados obtenidos fueron los siguientes:

R^2 : 0.8337732202287705
 MSE: 0.10366011037446508
 RSMSE: 0.32196290217114315
 MAE: 0.1533417855444603
 Error estandar: 0.3219642351177667

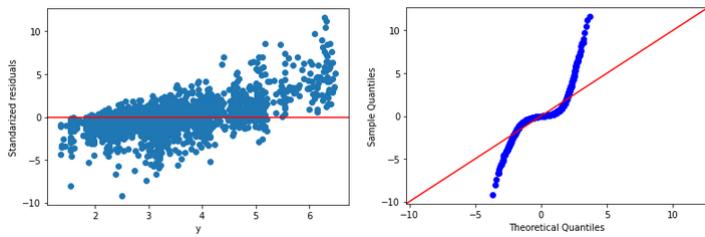


Gráfico de residuos y cuantil. (figuras 4.1.1 y 4.1.2)

Utilizaremos también los modelos Multi Layer Perceptron y Support Vector Regression para contrastarlos con los del paper.

Resultados del MLP:

R^2 : 0.8452408041827444
 MSE: 0.09650884978917687
 RSMSE: 0.31065873525329507
 MAE: 0.153458677001836
 Error estandar: 0.31070539268642416

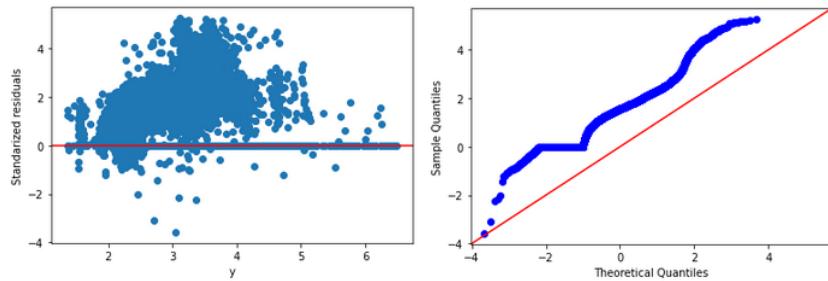


Gráfico de residuos y cuantil. (figuras 4.1.3 y 4.1.4)

Resultados del SVR:

R^2 : 0.8068711443896812
 MSE: 0.12043642135528387
 RSMSE: 0.34703950979000053
 MAE: 0.18047332110724765
 Error estandar: 0.3467160676989866

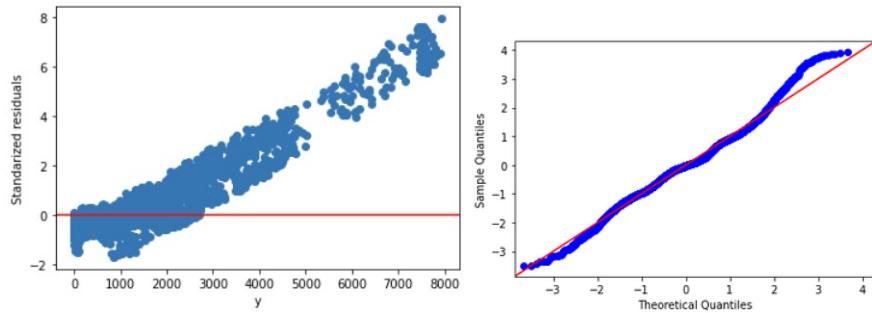


Gráfico de residuos y cuantil. (figuras 4.1.5 y 4.1.6)

Generaremos ahora un modelo regresión de tipo mínimos cuadrados ordinarios. Esto nos permitirá comparar este modelo con el anterior de Random Forest.

Resultados del OLS:
 R^2 : 0.5771135725862011
 MSE: 0.26371475042654247
 RSMSE: 0.5135316450098694
 MAE: 0.34110993914908105
 Error estandar: 0.5133338519175923

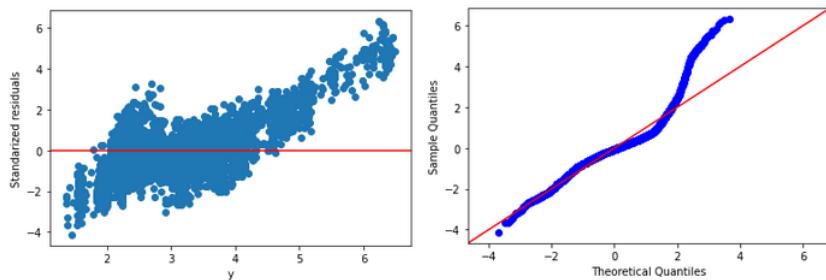


Gráfico de residuos y cuantil. (figuras 4.1.7 y 4.1.8)

4.2 Lasso y factor de inflación de varianza

Utilizaremos dos métodos para conocer saber si las variables independientes que elegimos tienen la suficiente influencia para quedarse en el modelo.

El primero será Lasso. Este método nos otorga los coeficientes de cada variable para ver cuál tiene más influencia que otras.

Los resultados obtenidos fueron los siguientes:

```
[ -1.25239704e-02  1.17108446e-02 -2.25235500e-03  8.22206662e-03
  1.42008591e-03  2.56597630e-03  1.54785389e-02 -1.22254590e-02
  2.03009591e-03  5.13341240e-03 -1.94424549e-03 -2.41175383e-03
 -1.52971933e-06  5.48523134e-07  2.47408363e-05 -6.51908366e-05
 -3.14883927e-06 -3.04336756e-05  7.14533728e-05]
2.170640546638908
```

Coefficientes obtenidos mediante LASSO. (figura 4.2.1)

El segundo método es con el factor de inflación de varianza. Este también nos muestra la influencia que tiene cada variable con la que queremos proyectar. En dado caso que una salga con un factor de inflación de varianza de más de 10, qué en general, un VIF superior a 10 indica una alta correlación y es motivo de preocupación, la descartaremos por tener una relación directa de la variable a proyectar. Lo cual generaría una desviación del modelo.

Los resultados obtenidos fueron los siguientes:

	features	vif_Factor
0	graySigma	3.057105e+01
1	hMean	6.263629e+01
2	vSigma	1.328840e+01
3	grayMean 0	2.819411e+01
4	vMean 0	1.174884e+02
5	hMean 1	7.661614e+01
6	hSigma 1	2.448502e+01
7	WeirPt1X	1.932004e+06
8	WeirPt1Y	8.425170e+05
9	WeirPt2X	9.543357e+06
10	WeirPt2Y	6.260092e+05
11	WwRawLineMin	4.169160e+00
12	WwRawLineMax	8.595448e+02
13	WwRawLineMean	2.169796e+04
14	WwRawLineSigma	5.015167e+03
15	WwCurveLineMax	7.891915e+02
16	WwCurveLineMean	2.138986e+04
17	WwCurveLineSigma	4.889393e+03

VIFs obtenidos. (figura 4.2.2)

Obtuvimos valores atípicos conforme a los siguientes métodos:

- Valores atípicos de residuos
- Valores altos de apalancamiento

- Valores altos de distancia de Cook
- Valores altos de DFFIT
- Valores altos de DFBetas

Una vez que obtuvimos estos valores, los eliminamos de nuestros datos de entrenamiento con el fin de ver si logramos generar mejores modelos. Habiendo eliminado los valores atípicos, generaremos los mismo modelos para ver si obtenemos una mejor efectividad.

Empezaremos generando primero un modelo con Random Forest por ser el que se utilizó en el Paper. Verémos si podemos generar una efectividad mayor a la mencionada por los investigadores.

Resultados del Random Forest trás eliminar datos atípicos :

R^2 : 0.8272486821654673
MSE: 0.1077288551141227
RSMSE: 0.3282207414441121
MAE: 0.15628777936281502
Error estandar: 0.32824741648132333

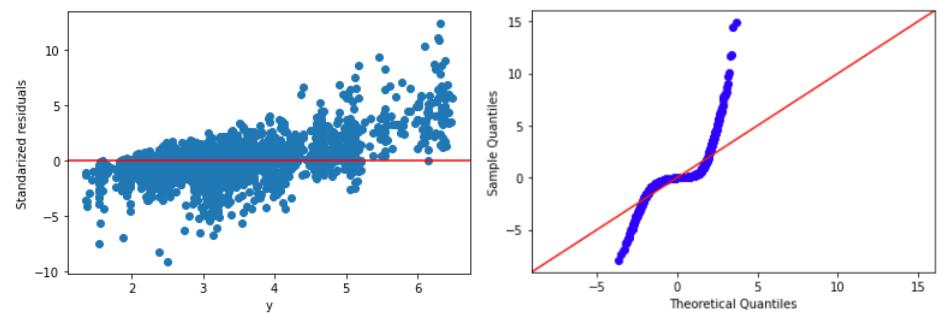


Gráfico de residuos y cuantil. (figuras 4.2.3 y 4.2.4)

Ahora, generaremos uno con MLP. Esto para contrastar su efectividad con la de Random Forest y comparar cuál es mejor.

Resultados del MLP trás eliminar datos atípicos :

R^2 : 0.8366963392403801
MSE: 0.10183723418208397
RSMSE: 0.3191194669431559
MAE: 0.1608734256261398
Error estandar: 0.31601780927805584

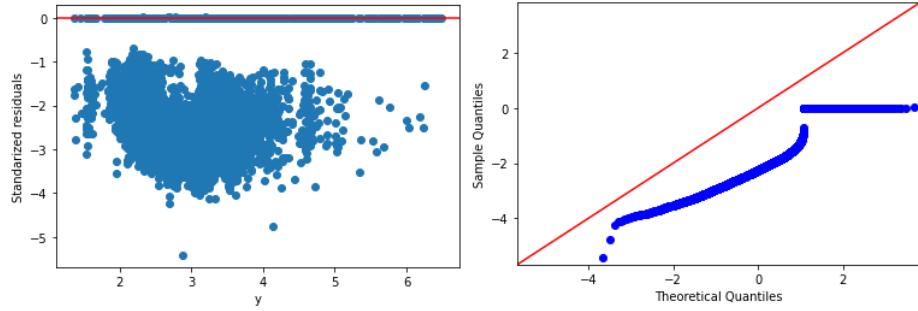


Gráfico de residuos y cuantil. (figuras 4.2.5 y 4.2.6)

Lo mismo con SVR:

R^2 : 0.8068711443896812
 MSE: 0.12043642135528387
 RSMSE: 0.34703950979000053
 MAE: 0.18047332110724765
 Error estandar: 0.3467160676989866

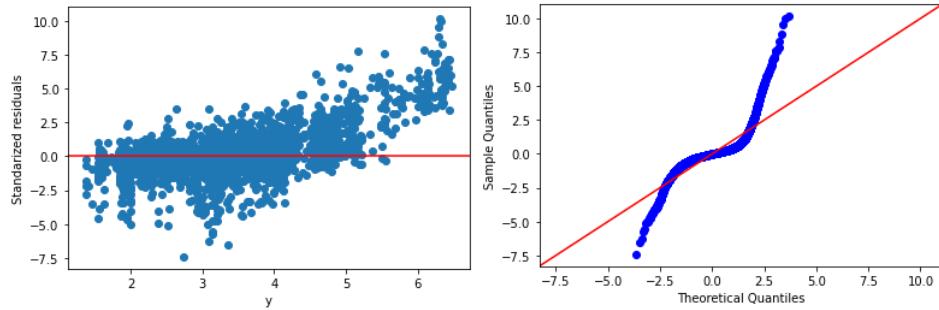


Gráfico de residuos y cuantil. (figuras 4.2.7 y 4.2.8)

Podemos comparar los resultados antes y después de retirar residuos observando la figura 4.2.9, en la cual indicamos el método y el método sin residuos con la expresión "s/r":

Method	R-squared	MSE	RSMSE	MAE	Std Error
RFR	0.83377	0.10366	0.32196	0.15334	0.32196
RFR - s/r	0.82724	0.10772	0.32822	0.15628	0.32824
MLP	0.84524	0.09650	0.31065	0.15345	0.31070
MLP - s/r	0.83669	0.10183	0.31911	0.16087	0.31601
SVR	0.80687	0.12043	0.34703	0.18047	0.34671
SVR - s/r	0.80687	0.12043	0.34703	0.18047	0.34671
OLS	0.57711	0.26371	0.51353	0.34110	0.51333

Tabla de comparación de métricas. (figura 4.2.9)

4.3 Discharge

Inicialmente generamos un modelo regresor con Random Forest sin limpiar los datos para comprobar la fiabilidad mostrada en la investigacion inicial. Los resultados obtenidos fueron los siguientes:

R^2 : 0.8247287083095377
 MSE: 233956.12149014932
 RSMSE: 483.690108943887
 MAE: 200.22409922729435
 Error estandar: 483.5868563381082

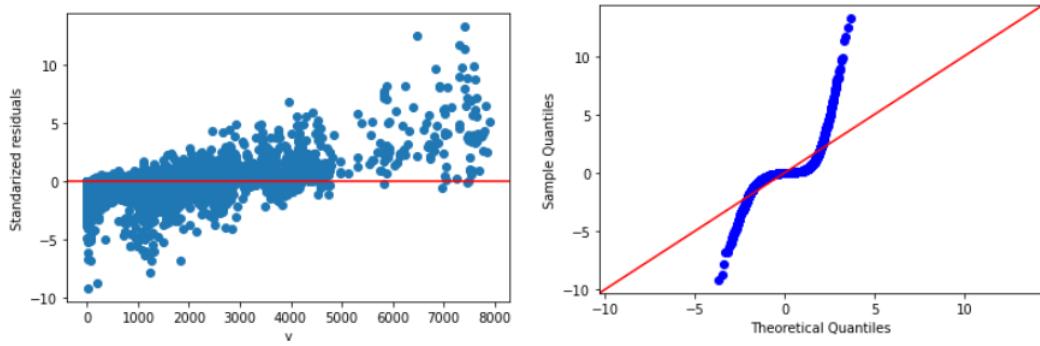


Gráfico de residuos y cuantil. (figuras 4.3.1 y 4.3.2)

Utilizaremos también los modelos MLP y SVR para contrastarlos con los del paper.

Resultados del MLP:

R^2 : 0.9106606310053236
 MSE: 119252.22929996347
 RSMSE: 345.3291608016379
 MAE: 157.9595011798805
 Error estandar: 345.3897105087386

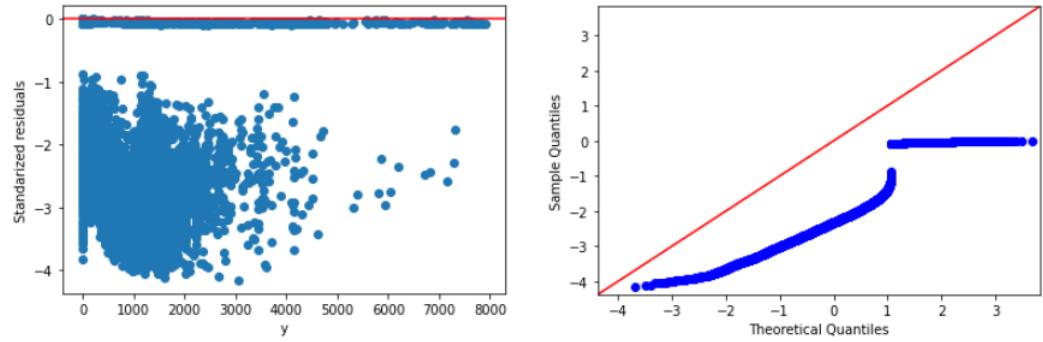


Gráfico de residuos y cuantil. (figuras 4.3.3 y 4.3.4)

Resultados del SVR:

R^2 : 0.47833182204956115

MSE: 696334.5933095505

RSMSE: 834.4666520056692

MAE: 370.20593703858617

Error estandar: 818.5175314021255

Los gráficos de residuos y cuantil lucen así:

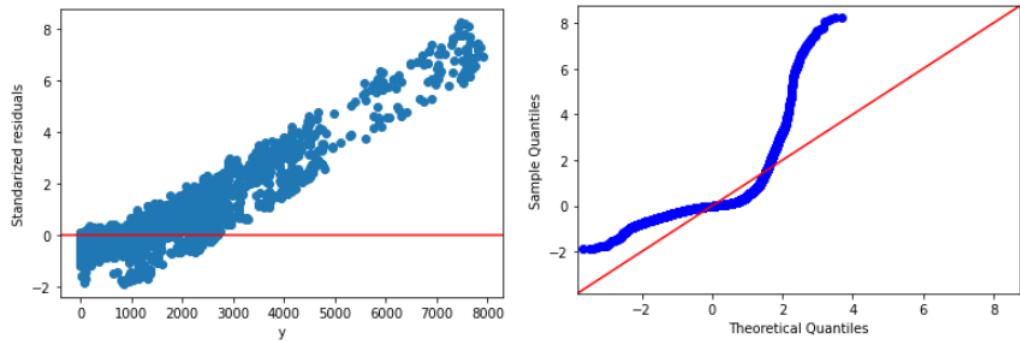


Gráfico de residuos y cuantil. (figuras 4.3.5 y 4.3.6)

Generaremos ahora un modelo regresión de tipo mínimos cuadrados ordinarios. Esto nos permitirá comparar este modelo con el anterior de Random Forest.

Resultados del OLS:

R^2 : 0.5825741299373127

MSE: 557189.5809496772

RSMSE: 746.4513252380741

MAE: 471.07220350721855

Error estandar: 746.029710625578

Los gráficos de residuos y cuantil lucen así:

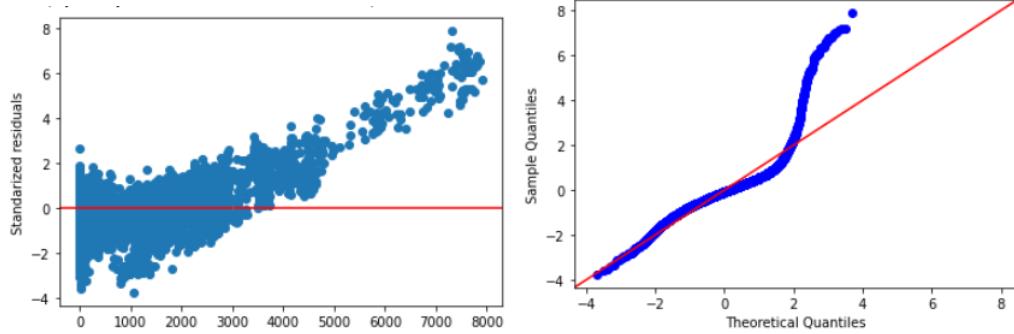


Gráfico de residuos y cuantil. (figuras 4.3.7 y 4.3.8)

4.4 Lasso y factor de inflación de varianza

Una vez más, utilizaremos LASSO y VIF para conocer saber si las variables independientes que elegimos tienen la suficiente influencia para quedarse en el modelo.

Los resultados obtenidos fueron los siguientes:

```
[ -9.46134860e+01  1.73567409e+02 -1.30637753e+04  7.62791997e+01
-1.48959685e+02 -4.23588021e+01  7.89276654e+01 -2.44886993e+01
5.98886936e+01  1.64691606e+01  3.26945918e+01  1.84438576e+03
-1.72173355e+01 -4.88554572e+01  8.88958750e+01  1.84314888e+00
5.10904651e+01  5.76515441e+01 -1.82703479e+00  1.01521820e+02
-7.56246079e+03  2.24644243e+03  1.57402750e+01 -1.08667974e+02
-4.33387073e+01 -1.46288750e+02 -2.33075285e+01 -7.41032630e+01
2.48466609e+00  4.68145566e+00 -2.18273840e+00 -1.97971625e+00
-6.64474554e-03 -7.28889608e-03  3.82315972e-02 -9.88586799e-02
-5.04423832e-03 -4.67612282e-02  1.17679214e-01]
777.4131998292414
```

Coeficientes obtenidos mediante LASSO. (figura 4.4.1)

	features	vif_Factor		
0	grayMean	8.058535e+04		
1	graySigma	2.066673e+04	21	graySigma 1 8.005780e+03
2	entropyMean	4.847876e+03	22	entropyMean 1 4.428803e+03
3	entropySigma	3.962144e+03	23	entropySigma 1 5.098589e+03
4	hMean	7.942564e+04	24	hMean 1 3.299515e+04
5	hSigma	2.076683e+04	25	hSigma 1 8.263393e+03
6	sMean	3.384433e+04	26	sMean 1 4.266018e+04
7	sSigma	4.865507e+01	27	sSigma 1 3.499547e+01
8	vMean	5.181360e+03	28	vMean 1 4.746214e+03
9	vSigma	6.192783e+01	29	vSigma 1 1.503577e+01
10	grayMean 0	1.492708e+04	30	WeirPt1X 2.181726e+06
11	graySigma 0	2.083634e+03	31	WeirPt1Y 9.441640e+05
12	entropyMean 0	1.475363e+03	32	WeirPt2X 1.077333e+07
13	entropySigma 0	1.380218e+03	33	WeirPt2Y 7.118787e+05
14	hMean 0	1.551499e+04	34	WwRawLineMin 4.248188e+00
15	hSigma 0	2.228781e+03	35	WwRawLineMax 9.295935e+02
16	sMean 0	3.072351e+04	36	WwRawLineMean 2.532234e+04
17	sSigma 0	3.787064e+01	37	WwCurveLineSigma 5.185550e+03
18	vMean 0	4.037329e+03	38	WwCurveLineMax 8.463212e+02
19	vSigma 0	1.520417e+01	39	WwCurveLineMean 2.494834e+04
20	grayMean 1	2.985415e+04	40	WwCurveLineSigma 5.068000e+03

VIFs obtenidos. (figura 4.4.2)

Una vez que obtuvimos estos valores, los eliminamos de nuestros datos de entrenamiento con el fin de ver si logramos generar mejores modelos. Habiendo eliminado los valores atípicos, generaremos los mismo modelos para ver si obtenemos una mejor efectividad.

Empezaremos generando primero un modelo con Random Forest por ser el que se utilizó en el Paper. Verémos si podemos generar una efectividad mayor a la mencionada por los investigadores.

Resultados del Random Forest trás eliminar datos atípicos :

R^2 : 0.823540678021367
 MSE: 235541.9314408406
 RSMSE: 485.32662346180905
 MAE: 200.50039029957205
 Error estandar: 485.2362009818441

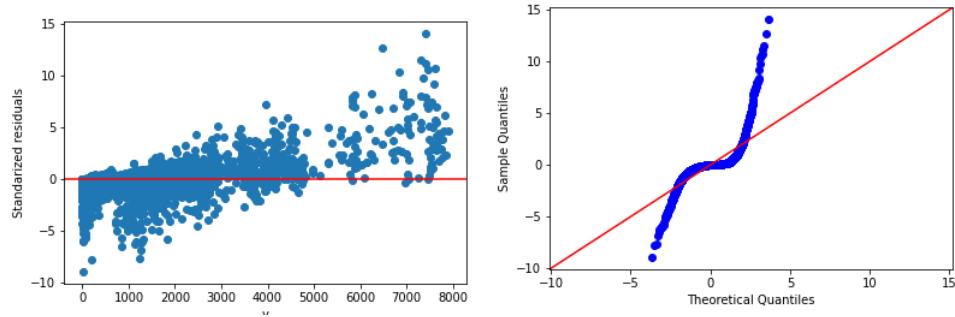


Gráfico de residuos y cuantil. (figuras 4.4.3 y 4.4.4)

Ahora, generaremos uno con MLP. Esto para contrastar su efectividad con la de Random Forest y comparar cuál es mejor.

Resultados del MLP trás eliminar datos atípicos :

R^2 : 0.9059160163932675
 MSE: 125585.44920092999
 RSMSE: 354.38037361136406
 MAE: 165.7657409533583
 Error estandar: 354.0705546392842

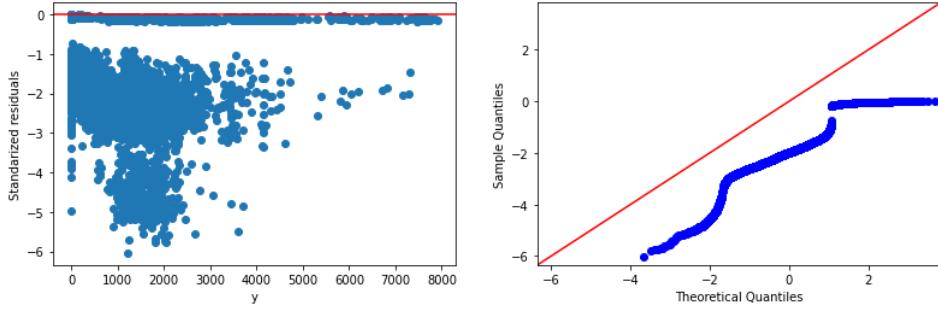


Gráfico de residuos y cuantil. (figuras 4.4.5 y 4.4.6)

Lo mismo con SVR:

R^2 : 0.47833182204956115
 MSE: 696334.5933095505
 RSMSE: 834.4666520056692
 MAE: 370.20593703858617
 Error estandar: 818.5175314021255

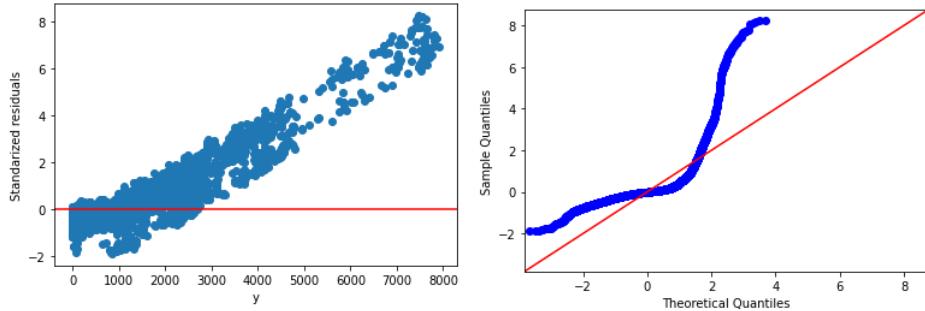


Gráfico de residuos y cuantil. (figuras 4.4.7 y 4.4.8)

Podemos comparar los resultados antes y después de retirar residuos observando la figura 4.2.9, en la cual indicamos el método y el método sin residuos con la expresión "s/r":

Method	R-squared	MSE	RSMSE	MAE	Std Error
RFR	0.82472	233956.1214	483.6901	200.2240	483.5868
RFR - s/r	0.82354	235541.9314	485.3266	200.5003	485.2362
MLP	0.91066	119252.2292	345.3291	157.9595	345.3897
MLP - s/r	0.90591	25585.4492	354.3803	165.7657	354.0705
SVR	0.47833	696334.5933	834.4666	370.2059	818.5175
SVR - s/r	0.47833	696334.5933	834.4666	370.2059	818.5175
OLS	0.58257	557189.5809	746.4513	471.0722	746.0297

Tabla de comparación de métricas. (figura 4.4.9)

4.5 Acercamiento

Tras analizar las figuras 4.2.9, 4.4.9 y observar con detalle nuestros resultados llegamos a dos puntos de interés:

Es más sencillo predecir el discharge de un cuerpo de agua que su stage, esto es una observación de utilidad para investigaciones futuras.

Si bien tenemos modelos que logran predecir de manera convincente tanto stage como discharge, el aporte que deseamos proveer con nuestro trabajo a la investigación previa tiene como intención no solo proveer modelos alternativos para la predicción sino también optimizar el proceso de entrenamiento e introducción de imágenes válidas a redes convolucionales.

5 Deep Learning aplicado a la problemática

Con la finalidad de realizar análisis óptimo de la alta cantidad de imágenes proporcionadas por el socio formador, como equipo optamos por un modelo de clasificación binaria aplicado a las imágenes, el modelo tiene como propósito principal, separar las imágenes optimas para análisis, de las imágenes con altas cantidades de ruido, ya sea obscuras, congeladas o inundadas.



Ejemplo de cuatro clases de imágenes encontrables (figura 5.0.1)

5.1 Entrenamiento y evaluación del modelo

Al igual que en la sección anterior del documento, dividimos nuestras imágenes en training, validation y testing, todo esto a la par que importamos las fotografías, también se definió el tamaño inicial y el batch size para el entrenamiento del modelo. La distribución de imágenes fué la siguiente:

group	images	classes
training	571	2
testing	580	2
validation	571	2

Distribución de las imágenes (figura 5.1.1)

Despues de haber importado las imagenes en estos grupos y con dos clases, vamos a empezar el desarrollo y entrenamiento de la red convolucional. Para este caso, probamos diferentes tipo de redes pre-entrenadas, por ejemplo: VGG16, VGG50, MobileNetV2, ResNet101V2, etc. Con la que obtuvimos un mejor accuracy fue con la última; pero no es la única capa que contiene la red. A continuación mostramos todas las capas y parámetros entrenables con las que cuenta:

```
Model: "sequential"
-----
Layer (type)          Output Shape       Param #
resnet101v2 (Functional)    (None, 16, 16, 2048)   42626560
conv2d (Conv2D)         (None, 14, 14, 64)      1179712
flatten (Flatten)       (None, 12544)        0
dense (Dense)          (None, 64)           802880
activation (Activation) (None, 64)           0
dropout (Dropout)       (None, 64)           0
dense_1 (Dense)         (None, 2)            130
activation_1 (Activation) (None, 2)           0
-----
Total params: 44,609,282
Trainable params: 44,511,618
Non-trainable params: 97,664
```

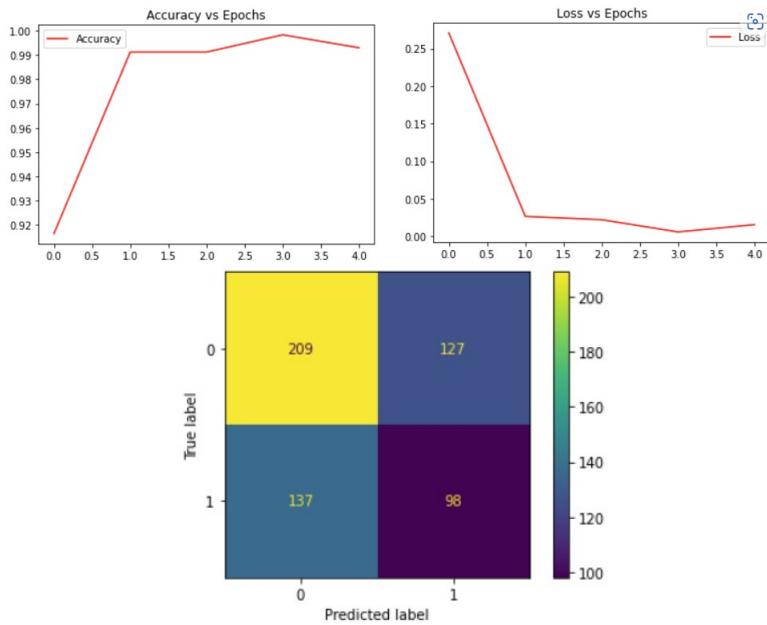
Descripción de la red convolucional (figura 5.1.1)

Esta red generó que para su entrenamiento tuvieramos que probar diferentes entornos, entre ellos: Collab, una instancia de AWS con el plan gratuito y nuestra propia computadora personal. El objetivo siempre fue que pudieramos probar el mejor modelo que encontraramos sin el obstaculo que pueden llegar a ser los recursos.

Con AWS definimos un tipo de instanciaEl tipo de instancia t2.micro. Nos ofrece una unidad SSD de 30 Gb de uso general de EBS, una sola CPU y una instalación limpia de Ubuntu 18.04 LTS. Seleccionamos este sistema operativo dado que es el entorno con el que más hemos trabajado a lo largo de la carrera. Esta instancia tiene un costo por servicio de 0.0116 USD por hora, lo cual es lo suficientemente bajo para poder trabajar con los 100 USD disponibles. Para

pasar las imágenes de nuestra computadora personal al servidor utilizamos la herramienta FTPS que ofrece Linux. A este servicio solamente se le define la carpeta de origen y de destino, el par de llaves de acceso a la instancia y la URL con puerto para hacer enlace en la conexión; y con eso pudimos hacer la transferencia del total de imágenes. Aunque, después de hacer toda la transferencia, nos acabamos el almacenamiento debido al peso de las imágenes y la cantidad de estas que utilizamos para el desarrollo.

Dado que nos terminamos el almacenamiento en AWS, recurrimos a utilizar Collab con el uso limitado de la GPU. Esto nos sirvió para disminuir considerablemente los tiempos de carga y entrenamiento. Para evaluar nuestro modelo de manera más precisa, generamos una matriz de confusión que nos indicara lo correcto o no que fue la predicción; siendo 1 las imágenes correctas y 0 las incorrectas. Al final pudimos obtener los siguientes resultados:



Graficos de resultados de la red (figura 5.1.3)

Train Loss: 0.0156 - Train Accuracy: 0.9929
Validation Loss: 0.5579 - Validation Accuracy: 0.9507
Test Loss: 0.003451 - Test Accuracy: 0.998248

5.2 Predicción con el modelo

Como vimos anteriormente, la matriz de confusión nos mostró que los resultados no fueron los esperados, pero es considerablemente eficiente encontrando imágenes incorrectas. Nos aprovechamos de eso y vamos a utilizar este modelo de clasificación para solamente filtrar las imágenes incorrectas. Haciendo

dicho lo anterior, generamos dos tipos de predicciones. Primero, buscaremos clasificar imagenes que sean de otra temporada diferente a las imagenes con las que entrenamos. Además, buscaremos después predecir imagenes que sean de la misma temporada, pero con 5 años de diferencia. De esta forma podremos comprobar nuestra hipotesis de ver la posibilidad de predecir algo en un lapso temporal pequeño y algo con un lapso temporal largo.

El dataset que generamos contienen las imagenes que nuestro modelo de clasificación puso como incorrectas:

	A	B	C
1	20170303StateLineWeir_20170303_Farrell_008.jpg		
2	20170303StateLineWeir_20170303_Farrell_010.jpg		
3	20170303StateLineWeir_20170304_Farrell_019.jpg		
4	20170303StateLineWeir_20170304_Farrell_033.jpg		
5	20170303StateLineWeir_20170304_Farrell_035.jpg		
6	20170303StateLineWeir_20170304_Farrell_036.jpg		
7	20170303StateLineWeir_20170304_Farrell_042.jpg		
8	20170303StateLineWeir_20170305_Farrell_045.jpg		
9	20170303StateLineWeir_20170305_Farrell_047.jpg		
10	20170303StateLineWeir_20170305_Farrell_065.jpg		
11	20170303StateLineWeir_20170306_Farrell_079.jpg		
12	20170303StateLineWeir_20170306_Farrell_083.jpg		
13	20170303StateLineWeir_20170306_Farrell_087.jpg		
14	20170303StateLineWeir_20170306_Farrell_089.jpg		
15	20170303StateLineWeir_20170306_Farrell_094.jpg		
16	20170303StateLineWeir_20170307_Farrell_109.jpg		
17	20170303StateLineWeir_20170307_Farrell_112.jpg		
18	20170303StateLineWeir_20170307_Farrell_113.jpg		
19	20170303StateLineWeir_20170308_Farrell_128.jpg		
20	20170303StateLineWeir_20170308_Farrell_130.jpg		
21	20170303StateLineWeir_20170308_Farrell_131.jpg		
22	20170303StateLineWeir_20170308_Farrell_140.jpg		
23	20170303StateLineWeir_20170308_Farrell_146.jpg		

Dataset resultante (figura 5.2.1)

Así, podremos leer las imagenes que queremos quitar y las removeremos del dataset de regresión. Para así verificar si el accuracy disminuyó o aumentó.

6 Resultados regresión con las imagenes clasificadas.

Enfocandonos de nuevo en el modelo de regresión, quitamos un total de 300 imagenes con la prediccción de diferentes epochas. Habiendo retirado estas imagenes, repetimos los procedimientos anteriores para verificar si nuestra hipotesis inicial era correcta o no. Para este nuevo caso vamos a probar un procedimiento diferente. Dado que vimos con las correlaciones que el stage y el discharge tienen una correlación de 0.97, vamos a intentar predecir uno y utilizar el resultado para predecir el otro. Vimos que tuvimos más efectividad prediciendo el discharge por lo que predeciremos este primero, y luego utilizarémos la prediccción para predecir el stage.

6.1 Discharge

6.1.1 Dos temporadas diferentes, mismo año

Resultados del Random Forest:

R^2 : 0.8428585325677589
MSE: 230830.60878807204
RSMSE: 480.4483414354472
MAE: 199.334044234427
Error estandar: 480.519141246104

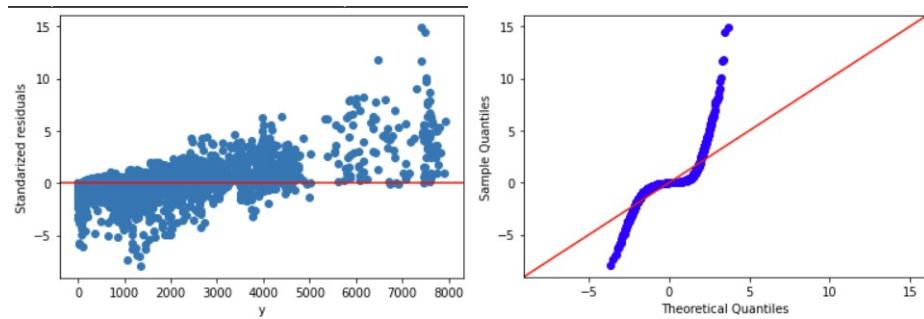


Gráfico de residuos y cuantil. (figuras 6.1.1 y 6.1.2)

Ahora, generaremos uno con MLP. Esto para contrastar su efectividad con la de Random Forest y comparar cuál es mejor.

Resultados del MLP:

R^2 : 0.9157598224645278
MSE: 123743.27771192083
RSMSE: 351.77162721277114
MAE: 160.9001891478277
Error estandar: 350.27568388863943

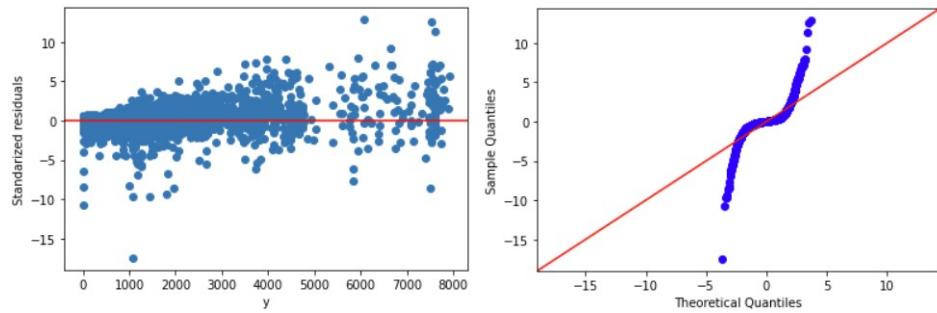


Gráfico de residuos y cuantil. (figuras 6.1.3 y 6.1.4)

Resultados del SVR:

R^2 : 0.4464374257558764
MSE: 813147.4658028209
RSMSE: 901.7468967525316
MAE: 390.64895141446806
Error estandar: 881.2995623089982

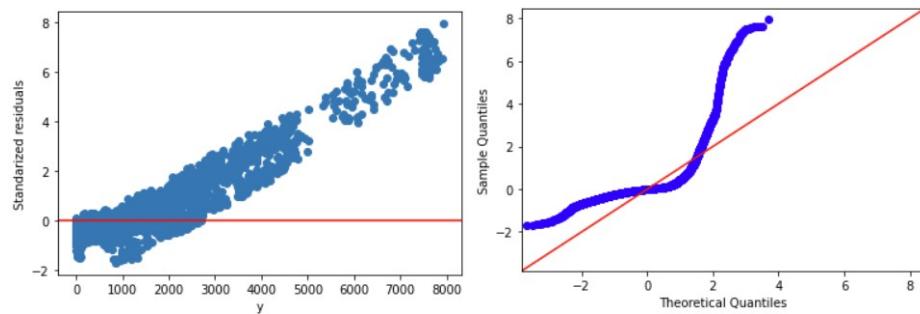


Gráfico de residuos y cuantil. (figuras 4.4.7 y 4.4.8)

6.1.2 Misma temporada, 5 años de diferencia

Resultados del Random Forest:

R^2 : 0.8399852532461007
MSE: 235051.26948237605
RSMSE: 484.8208632911501
MAE: 200.801479362815
Error estandar: 484.8879655345423

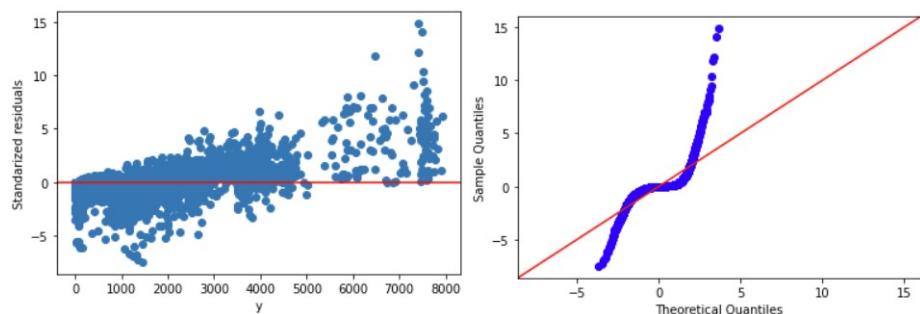


Gráfico de residuos y cuantil. (figuras 6.1.1 y 6.1.2)

Resultados del MLP:

R^2 : 0.9110624489392322
 MSE: 130643.48571346734
 RSMSE: 361.4463801360685
 MAE: 162.61324691171185
 Error estandar: 361.17446747102395

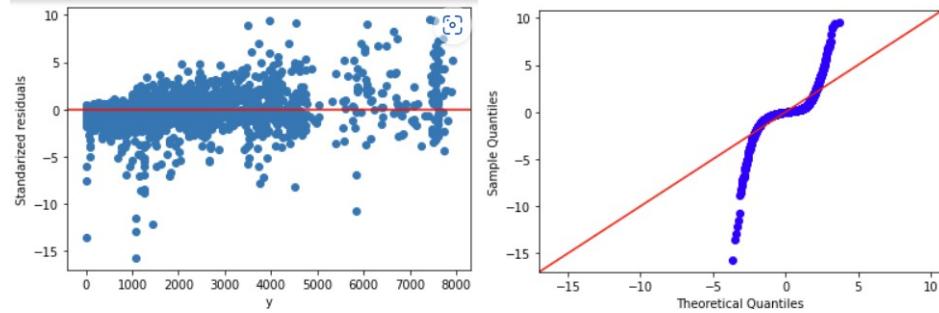


Gráfico de residuos y cuantil. (figuras 6.1.3 y 6.1.4)

Resultados del SVR:

R^2 : 0.4464374257558764
 MSE: 813147.4658028209
 RSMSE: 901.7468967525316
 MAE: 390.64895141446806
 Error estandar: 881.2995623089982

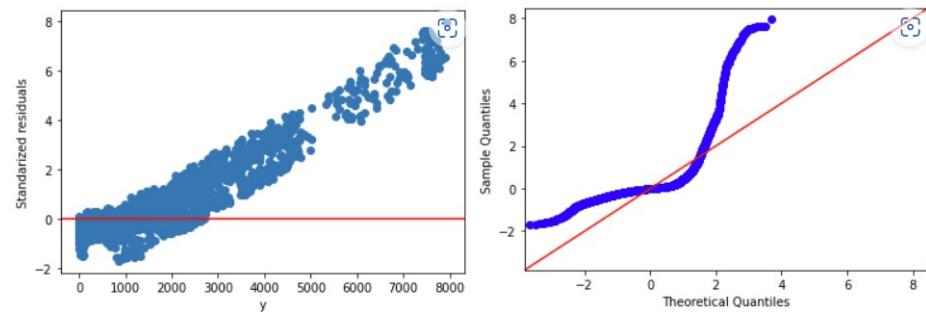


Gráfico de residuos y cuantil. (figuras 6.1.5 y 6.1.6)

6.2 Stage

Habiendo predicho el discharge, vamos a pasar las predicciones como parámetro a Stage. De esta forma deberíamos obtener una mejor predicción.

6.2.1 Dos temporadas diferentes, mismo año

Resultados del Random Forest:

R^2 : 0.9348233685274212
 MSE: 0.043348343158582964
 RSMSE: 0.20820264925928048
 MAE: 0.10233146695197334
 Error estandar: 0.20823741404732551

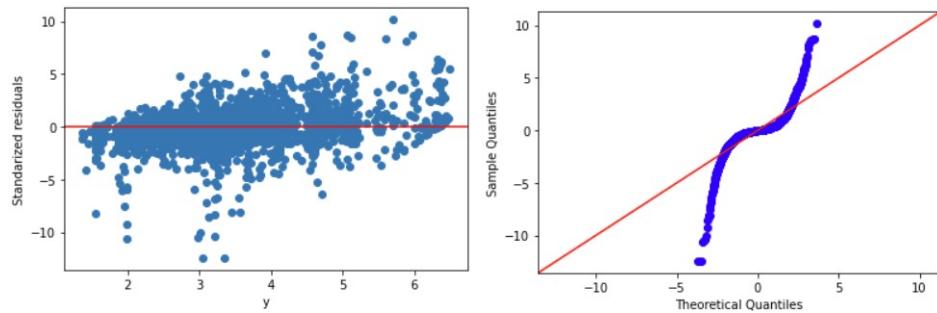


Gráfico de residuos y cuantil. (figuras 6.2.1 y 6.2.2)

Resultados del MLP:

R^2 : 0.9289439205480987
 MSE: 0.04725870677867795
 RSMSE: 0.21739067776396934
 MAE: 0.12115135168475165
 Error estandar: 0.21460197784806434

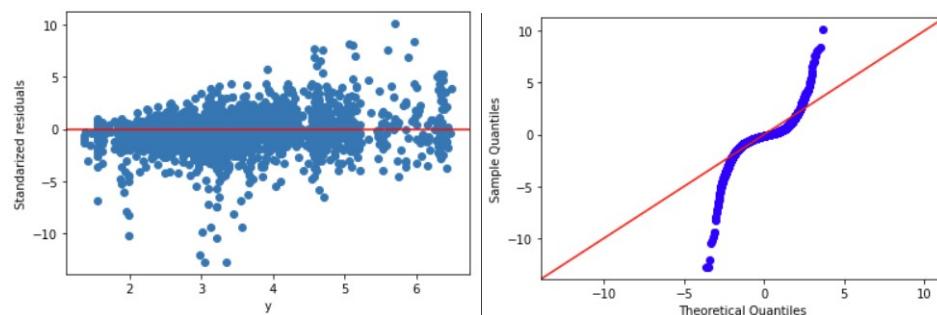


Gráfico de residuos y cuantil. (figuras 6.2.3 y 6.2.4)

Resultados del SVR:

R^2 : 0.9340041135981467
 MSE: 0.04389322163735938
 RSMSE: 0.20950709209322577
 MAE: 0.11508667649868728
 Error estandar: 0.20952219080878662

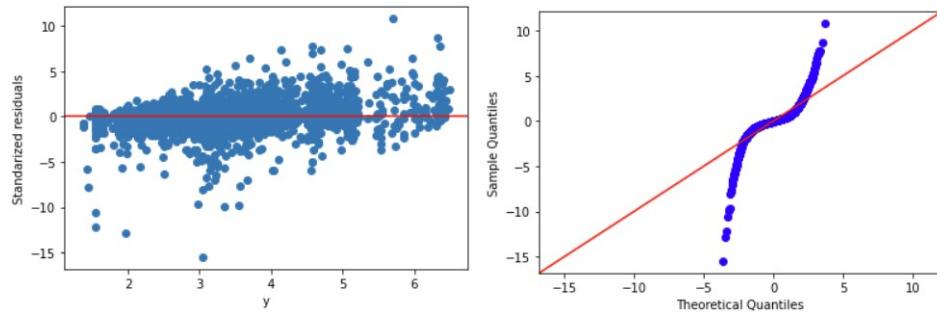


Gráfico de residuos y cuantil. (figuras 6.2.5 y 6.2.6)

6.2.2 Misma temporada, 5 años de diferencia

Resultados del Random Forest:

R^2 : 0.9347624280044261
 MSE: 0.043388874107227755
 RSMSE: 0.2082999618512393
 MAE: 0.10250343556823586
 Error estandar: 0.20833654038438645

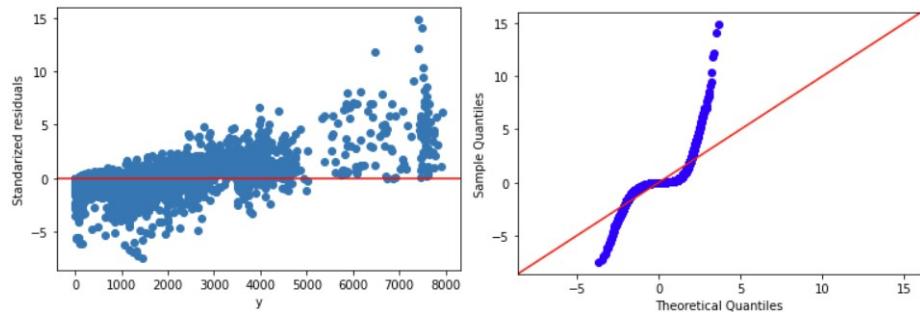


Gráfico de residuos y cuantil. (figuras 6.1.1 y 6.1.2)

Resultados del MLP:

R^2 : 0.9307376890618712
 MSE: 0.046065688800834406

RSMSE: 0.21462918906997344
 MAE: 0.11710421955192502
 Error estandar: 0.2143773137454902

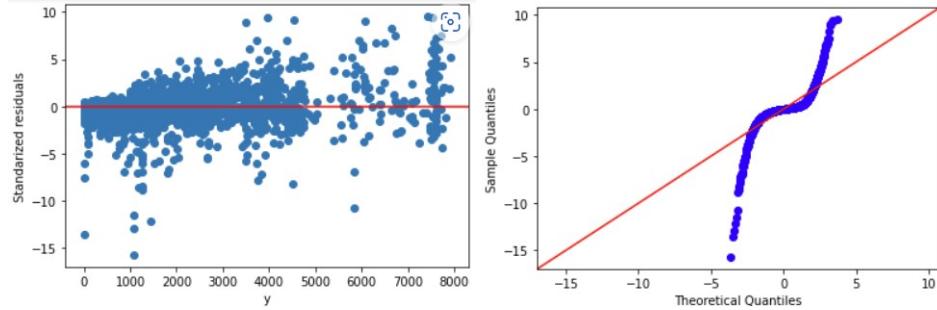


Gráfico de residuos y cuantil. (figuras 6.1.3 y 6.1.4)

Resultados del SVR:

R^2 : 0.9340041135981467
 MSE: 0.04389322163735938
 RSMSE: 0.20950709209322577
 MAE: 0.11508667649868728
 Error estandar: 0.20952219080878662

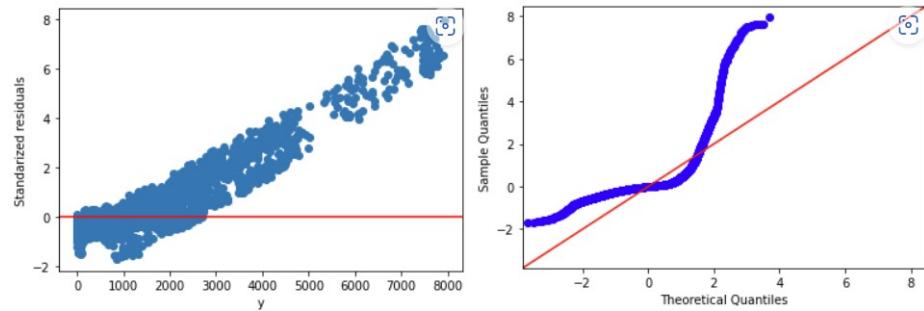


Gráfico de residuos y cuantil. (figuras 4.4.7 y 4.4.8)

7 Conclusiones Finales

Despues de ver los resultados, creemos que la hipotesis inicial se cumplió en uno de los ámbitos especificados.

Primero, pudimos hacer un modelo de predicción con 0.934 de efectividad para el stage y 0.911 para el discharge. Este no se cumplió de la forma que queríamos ya que no llegamos al 0.95 buscado para poder tomar la hipotesis como verdadera. Además, notamos una muy pequeña diferencia en efectividad

al hacer la eliminación de las imágenes. Aunque también cabe destacar que solo hicimos eliminación de imágenes por temporadas, no clasificamos el dataset entero.

Segundo, sí hay una diferencia detectable entre la predicción de imágenes en temporadas diferentes y temporadas iguales. Esto nos dice que sí hay una afectación o cambio en las imágenes que se clasifican conforme se cambia de estación. Esto puede ser muy útil para generar investigación a futuro que nos ayude a predecir el cambio en las estaciones por el cambio climático.

Al final, a pesar de no poder generar un modelo que alcanzara 0.95 de éxito, creemos que hemos aportado un avance importante para la investigación y el análisis de este río. Se puede tomar el desarrollo previamente realizado para enriquecer las pruebas y el estudio de las diferentes variables que envuelven al río. En general, confiamos y esperamos que esta investigación nos acerque al desarrollo de un sistema que envuelva más objetos de agua, no solo ríos. Para que así, podamos tener un análisis profundo de las condiciones y características de los cuerpos de agua que se extienden por México.