ISSN: 1692-7257 - Volumen 2 – Número 42 - 2023

Revista Colombiana de Tecnologías de Avanzada

Recibido: 10 de abril 2023
Aceptado: 15 de julio 2023

PREDICCIÓN DEL FENÓMENO DE LA PRECIPITACIÓN AMBIENTAL EN EL MUNICIPIO DE AQUITANIA

PREDICTION OF THE ENVIRONMENTAL PRECIPITATION PHENOMENON IN AQUITANIA

Ing. Viviana M. Bernal-Benítez[*], Ing. Juan C. Gómez-Malagón[*], MSc. Camilo Pardo-Beainy[*]

[*] **Universidad Santo Tomás,** Electronic Engineering Faculty.
Av. Universitaria No. 45 – 202, Tunja, Boyacá Colombia.
Tel.: 57-8-7440404, Ext. 5612
E-mail: {viviana.bernal, juan.gomezm, camilo.pardo}@usantoto.edu.co.

Cómo citar: Bernal-Benítez, V. M., Gómez-Malagón, J. C., & Pardo-Beainy, C. (2023). PREDICCIÓN DEL FENÓMENO DE LA PRECIPITACIÓN AMBIENTAL EN EL MUNICIPIO DE AQUITANIA. REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA), 2(42), 17–22. https://doi.org/10.24054/rcta.v2i42.2649

Resumen: La ciencia de la meteorología genera importantes predicciones sobre los fenómenos que ocurren día a día en la atmósfera y que son de gran importancia para las actividades humanas como la agricultura, la sostenibilidad de los ecosistemas y el análisis climático. Con este Proyecto, se busca crear un sistema predictivo de precipitación atmosférica que trabaja con técnicas de Machine Learning haciendo uso de datos recolectados del monitoreo climático sobre el municipio de Aquitania en el departamento de Boyacá. Para generar este sistema predictivo de precipitación atmosférica, se utilizan los recursos de IBM Watson y la herramienta para la creación de código en Python: Jupyter Notebook. El algoritmo es entrenado empleando un conjunto de datos que contiene 35 años de información meteorológica tomados de la vereda Hoya La Manzana. El proceso desarrollado inicia con el refinamiento y limpieza del conjunto de datos, a continuación, la creación del modelo de entrenamiento con el 80% del dataset para proceder con la prueba del algoritmo empleando el 20% restante y finaliza con el análisis de los resultados obtenidos en la implementación del sistema predictivo apoyándose en métricas de evaluación tales como precisión, exactitud, sensibilidad del sistema, las cuales permiten observar las variaciones en el desempeño de cada uno de los modelos. Se consiguió una precisión de casi el 96% con el algoritmo fundamentado en Arboles de decisión, siendo este un posible punto de partida para la construcción de una herramienta de alta eficiencia que permita a los agricultores aumentar la productividad de la tierra, anticipándose a los posibles cambios climáticos que puedan afectar la salud y el desarrollo de sus cultivos.

Palabras Clave: Aumento de Gradiente, Precipitación Ambiental, Python, IBM Watson, Meteorología, Aprendizaje automático.

Abstract: The science of meteorology generates important predictions about the phenomena, which occur in the atmosphere every day and have a great importance in human activities such as agriculture, the sustainability of ecosystems and climate analysis. This project seeks to create a predictive system for atmospheric precipitation, which works with Machine Learning techniques using data collected from climate monitoring over Aquitania, a town in Boyacá department. To generate this classifier algorithm, the resources of IBM Watson and the tool to create the code in Python: Jupyter Notebook. The algorithm is trained using a dataset, which contains 35 years of meteorological information taken from the settlement Hoya La Manzana. The process developed begins with the refinement and cleaning of the dataset, then, the creation of the training model with 80% of the dataset to proceed with the algorithm test using the remaining 20% and finishes with the analysis of the results obtained in the predictive system implementation relying on evaluation metrics such as precision, accuracy, sensitivity of the system, which allow identifying the variations in performance of each model. An accuracy of almost 96% was achieved with the algorithm based on Decision Trees, this being a possible starting point for the construction of a high-efficiency tool that allows farmers to increase the productivity of the land, anticipating possible climatic changes, which may affect their health and the development of their crops.

Keywords: Gradient Boost, Environmental Precipitation, Python, IBM Watson, Meteorology, Machine Learning.

1. INTRODUCCIÓN

La meteorología es una disciplina que estudia y predice los diferentes fenómenos que ocurren en la atmósfera (Wardah et al., 2008), y proporciona importantes predicciones diarias útiles para diferentes actividades humanas como: la agricultura (Lu et al., 2017), aeronáutica (Kok et al., 2015), navegación (Bosy et al., 2010), actividades militares (Rozenstein & Karnieli, 2011), predicción de enfermedades (Ma et al., 2020) y prevención de incendios (Gonçalves et al., 2006). Las estaciones meteorológicas están formadas por instrumentos que miden, registran y comparten información sobre diversos factores como: temperatura, humedad, presión atmosférica, etc., y, posteriormente, realizan registros y los comparten con otras estaciones (Colston et al., 2018). Para generar previsiones atmosféricas precisas, es importante disponer de un número considerable de estaciones distribuidas por todo el territorio. Cuando se consulta la base de datos IDEAM (Sotelo et al., 2020), se evidencia que en el departamento de Boyacá muchas estaciones no están en funcionamiento o no suministran el 100% de la información necesaria para crear un modelo que alerte a las poblaciones propensas a sufrir catástrofes ambientales. Por esta razón, esta investigación se orienta al diseño de un algoritmo de predicción que permita saber si lloverá o no, utilizando un conjunto de datos que contiene 35 años de información meteorológica sobre radiación solar, temperatura y humedad relativa sobre el asentamiento La Hoya en Aquitania, Boyacá.

A continuación, se presentan algunas referencias al estado de la técnica relacionado con los sistemas predictivos para el análisis de datos meteorológicos:

F. Riabani., W. García. y J. Herrera. (Riabani Mercado et al., 2016) implementaron una red neuronal entrenada con el algoritmo propuesto por Huang para predecir heladas meteorológicas en el departamento de Cochabamba en Bolivia.

En China, G. Chen., S. Li. Y L. Knibbs. (Chen et al., 2018), utilizando datos sobre el espesor óptico de los aerosoles, la meteorología y otros factores de predicción, elaboraron un modelo de bosque aleatorio y dos modelos de regresión tradicionales para estimar las concentraciones de PM2,5 a nivel del suelo.

H. Han., S. Lee. y M. Kim. (Han et al., 2015) crearon un algoritmo oficial para la detección de la iniciación de nubes convectivas sobre el noreste de Asia, utilizando información del generador de imágenes meteorológicas y datos del satélite COMS1. Se basaron en tres enfoques de aprendizaje automático: árboles de decisión, bosques aleatorios y máquinas de vectores de apoyo con el fin de mitigar los daños causados por los peligros de iniciación convectiva.

2. MATERIALES Y MÉTODOS

Los recursos de IBM Watson y la codificación en Python se utilizan para crear el algoritmo clasificador con el fin de comparar el rendimiento de estos dos servicios.

2.1 Algoritmo clasificador mediante IBM Watson

IBM Watson es una plataforma de Inteligencia Artificial con una colección de servicios y habilidades que incluye Machine Learning. La Figura 1 presenta el esquema general de implementación; para el caso de estudio, se utilizó un algoritmo clasificador Gradient Augmentation sobre la aplicación generada llamada Rain Alert.

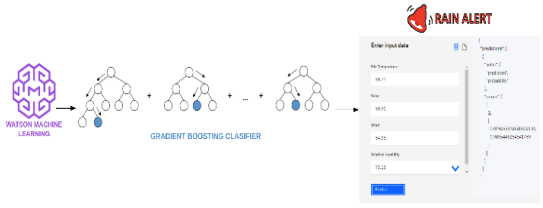


Fig. 1. Esquema general, aplicación del algoritmo con IBM Watson

Inicialmente, con el servicio IBM Watson Studio, se refina el conjunto de datos original y se normalizan todos los datos. Esto permite que los atributos tengan el mismo peso cuando se crea el algoritmo. A continuación, con el servicio AutoAI, se construye e implementa el modelo de aprendizaje. Realiza automáticamente la selección de los modelos que mejor se ajustan a los datos. El usuario de este servicio debe proporcionar un conjunto de datos en formato .csv e indicar el atributo a predecir y, a continuación, desplegar el modelo seleccionado. Opcionalmente, el usuario puede intervenir en el proceso de AutoAI durante la selección de: la métrica de optimización, los algoritmos a ejecutar, la cantidad de datos asignados para entrenamiento y prueba, así como, el valor asociado al resultado positivo. Tras aplicar las modificaciones anteriores, se crea el modelo de entrenamiento y se muestra el modelo de construcción para cada canalización, tal y como se muestra en la Figura 2.

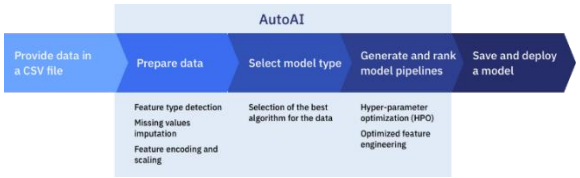


Fig. 2. Proceso AutoAI. (IBM's AutoAI at Work: Dos aplicaciones reales | por Álvaro Corrales Cano | Garaje IBM | Medio, n.d.)

Al final de este paso, se crea una tabla (Fig. 3) en la que se clasifica cada cadena de procesos en función de la precisión de la métrica de evaluación. Como puede observarse, el algoritmo que mejor se ajusta al conjunto de datos es el clasificador de refuerzo de gradiente.

Clasific... ↑	Nombre	Algoritmo	Precisión (Optimiza...	Mejoras
★ 1	3 interconexi...	Clasificador de aumento de gradiente	0.927	HPO-1 FE
2	4 interconexi...	Clasificador de aumento de gradiente	0.927	HPO-1 FE HPO-2
3	1 interconexi...	Clasificador de aumento de gradiente	0.926	*Ninguno*
4	2 interconexi...	Clasificador de aumento de gradiente	0.926	HPO-1
5	5 interconexi...	Clasificador de árboles adicionales	0.917	*Ninguno*
6	6 interconexi...	Clasificador de árboles adicionales	0.917	HPO-1
7	7 interconexi...	Clasificador de árboles adicionales	0.917	HPO-1 FE
8	8 interconexi...	Clasificador de árboles adicionales	0.917	HPO-1 FE HPO-2

Fig. 3. Clasificación de los conductos generados

Este algoritmo clasificador de refuerzo de gradiente se utiliza para la clasificación y la regresión. Se basa en la combinación de modelos predictivos débiles, que suelen ser Árboles de Decisión, para crear un modelo predictivo fuerte. La generación de árboles de decisión débiles se realiza de forma secuencial, cada árbol se crea de tal forma que corrige los errores del árbol anterior (Natekin & Knoll, 2013). Uno de los parámetros de este tipo de argumentos es la tasa de aprendizaje que controla el grado de mejora del árbol respecto al anterior.

La ecuación (1) (Friedman, 2001) es la expresión matemática de este algoritmo, donde $\hat{y}$ es la predicción a optimizar. Para ello se crea una función objetivo basada en el error cuadrático que hay que minimizar.

$$obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) \qquad (1)$$

Teniendo en cuenta todas las ventajas de este algoritmo clasificador, se implementa en Jupyter Notebook siguiendo la topología de scikit-learn.

2.2 Algoritmo clasificador en Python

En los scripts de Jupyter Notebook, se replica el algoritmo clasificador de incremento de gradiente y también se implementa el algoritmo de árbol de decisión simple con el fin de realizar una comparación entre ambos utilizando las diferentes métricas de evaluación. Para este conjunto de datos, se realiza una limpieza y organización previa de los datos, eliminando algunas variables como fecha, latitud y longitud. Con las variables de temperaturas máximas y mínimas de cada día se calcula un valor medio que se almacena en la variable 'Min Termperature' y se define también la variable 'Precipitation' como parámetro a clasificar. La figura 4 muestra una sección del conjunto de datos modificado.

Min Temperature	Solar	Wind	Relative Humidity	Precipitation
12.2055	16.043057	2.128112	0.820475	True
12.6995	23.784239	2.249125	0.784151	False
12.5330	26.742218	2.205755	0.738471	False
11.5445	23.773327	2.110489	0.752569	True
11.7230	27.595187	2.501989	0.619903	False

Fig. 4. Encabezado del conjunto de datos modificado

Con esta información es posible separar las variables independientes (Min Temperature, Solar, Wind, Relative Humidity) de la variable dependiente (Precipitation). Además, el 80% del conjunto de datos se define para el entrenamiento del algoritmo y el 20% para la prueba del algoritmo. El proceso de esta división de datos se ilustra en la Figura 5.

```
#Se importa el dataset arreglado y normalizado
df = pd.read_csv("NuevoClima3.csv")
cdf = df.iloc[0:,1:6]
#se asigna la variable precipitacion, como la variable independiente
X = cdf.iloc[0:,0:4]
Y = cdf.Precipitation
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
X.head()
#Tamaño de los datos de entrenamiento y los datos de prueba
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

Fig. 5. División de datos

Utilizando las distintas herramientas de la biblioteca scikit-learn, se llaman las funciones 'DecisionTreeClassifier' y 'XGBClassifier' para entrenar los modelos y se utiliza el comando "predict" para probarlos (Fig. 6).

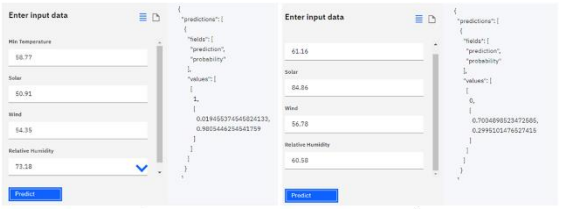
```
#Arboles de decisión
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
algoritmo1 = DecisionTreeClassifier(criterion = 'entropy')
algoritmo=XGBClassifier()
#Entrenamiento el modelo
algoritmo.fit(X_train, y_train)
algoritmo1.fit(X_train, y_train)
#predicción
y_pred = algoritmo.predict(X_test)
y_met=algoritmo.score(X_train,y_train)
y_pred1 = algoritmo1.predict(X_test)
y_met1=algoritmo1.score(X_train,y_train)
```

Fig. 6. Entrenamiento y prueba de algoritmos

3. RESULTADOS

3.1 Algoritmo de clasificación con IBM Watson

Una vez seleccionado el algoritmo que mejor se ajusta a los requisitos, se despliega y se prueba, como se muestra en la figura 7a. A la izquierda se introduce la información correspondiente a las variables independientes y, al pulsar sobre la casilla de predicción, se aplica el algoritmo de clasificación y se obtiene el resultado de la derecha. En el caso de la figura 7a, se obtiene una respuesta positiva (Valores=1), lo que indica que con estos parámetros existe un 98% de probabilidad de precipitación. En contraste con el resultado obtenido en la figura 7b, donde hay un 70% de probabilidad de que no llueva.



a b

Fig. 7. Sistema de predicción de precipitaciones (a) Resultado positivo (b) Resultado negativo.

Como puede observarse, en el gráfico de la curva ROC (Fig. 8), la relación entre la proporción de resultados positivos correctamente predichos frente a la proporción de resultados negativos falsamente predichos como positivos, indica un alto nivel de rendimiento del modelo y queda corroborado por el valor del área bajo la curva, que se refiere al 95,7% de predicciones correctas.

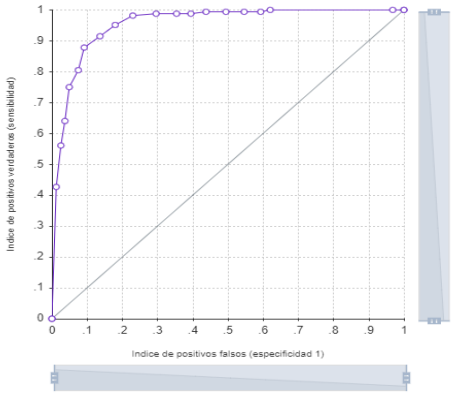


Fig. 8. Curva ROC

Además, se determinaron algunas métricas de evaluación del algoritmo para corroborar el rendimiento (Tabla 1).

Tabla 1: Métricas de evaluación del algoritmo.

Métrica de evaluación	Puntaje
Precisión	0.927
Recall	0.970
ROC AUC	0.957
F1	0.959

3.2 Algoritmo clasificador en Python

Para cada algoritmo se generó la matriz de confusión, que permite comparar la cantidad de datos que fueron clasificados correctamente y también observar las variaciones de cada una de las

métricas de evaluación de los modelos. Las Figuras 9 y 10 presentan las matrices de confusión de los algoritmos XG Boost y Árbol de Decisión.

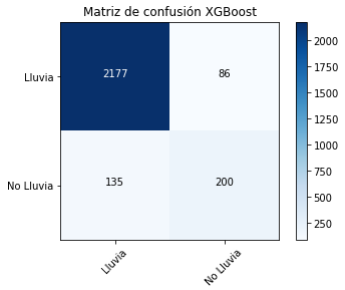


Fig. 9. Matriz de confusión para XG Boost

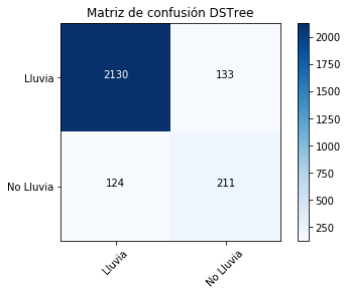


Fig. 10. Matriz de confusión para el árbol de decisión.

Para el algoritmo clasificador desarrollado utilizando Python, se calcularon las métricas de evaluación de cada modelo utilizando la librería 'sklearn.metrics', a partir de las cuales se obtuvieron los resultados presentados en la Tabla 2.

Table 2: Algorithm evaluation metrics

Métrica de evaluación	XG Boost	DS Árbol
Precisión	0.941609	0.944987
Exactitud	0.914935	0.901078
Sensibilidad	0.961997	0.941228
F1 Puntaje	0.951694	0.943104

Utilizando Matplotlib, algunas de estas métricas pueden visualizarse gráficamente. La curva ROC (Fig. 11) representa la relación entre la sensibilidad y la tasa de falsos positivos.

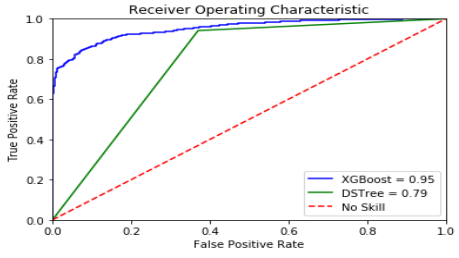


Fig. 11. Curva ROC.

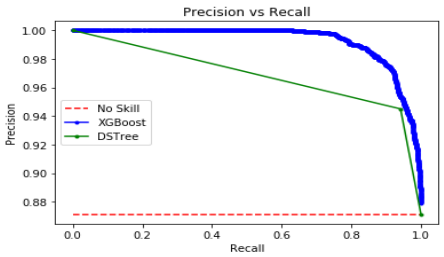


Fig. 12. Curva PR.

También se realizó el gráfico de la relación entre la precisión y la sensibilidad de los modelos (Fig. 12). Se puede decir que estas dos métricas son inversamente proporcionales, si el modelo es muy sensible y predice muchos datos como lluvia, el error aumentará. Sin embargo, si la precisión se incrementa cerca del 100%, el modelo aumentará la cantidad de Falsos Negativos, lo que a su vez disminuiría el recall. El objetivo de este gráfico es que esté lo más cerca posible de la esquina superior derecha, donde ambas variables tienen su valor máximo.

4. CONCLUSIONES

En ambos algoritmos, el desequilibrio de los datos es evidente, por lo que la curva ROC no resulta muy útil. En este caso, la curva PR proporciona información más asertiva sobre la precisión del modelo. En el caso del clasificador XG Boost, se obtiene una mayor precisión y una curva más suave que la del árbol de decisión simple. Se debe a la variabilidad en el porcentaje de probabilidad de que un dato predicho sea 1 o 0, que en el caso del árbol de decisión es una constante (0,94).

Con el algoritmo basado en Árboles de Decisión se alcanzó una precisión cercana al 96%. Siendo este, un posible punto de partida para la construcción de una herramienta altamente eficiente que permita a los agricultores aumentar la productividad de la tierra, anticipándose a posibles cambios climáticos que puedan afectar a la salud y el trabajo en sus cultivos.

El algoritmo clasificador de aumento de gradiente genera una disminución de los falsos negativos, pero un aumento de los falsos positivos, como se ve en la figura 12. Los dos modelos tienen una precisión cercana, pero el aumento de gradiente proporciona una mayor sensibilidad.

Utilizar la herramienta de aprendizaje automático de Watson Studio supuso una gran ventaja para identificar el algoritmo que mejor se ajustaba a las características del conjunto de datos. Además, es un servicio accesible y fácil de usar, lo que permite que

trabajadores de campo o entidades gubernamentales locales, específicamente de Aquitania, puedan utilizarlo. Este es un gran paso adelante en la apropiación de la tecnología en los procesos agrícolas.

Por último, cabe aclarar que, si bien es un tema importante, llevaría tiempo desarrollarlo a gran escala. Esto representa una invitación para que este tipo de investigaciones continúen e incorporen información diversa como: qué es más apropiado sembrar dependiendo de las condiciones climáticas, registros de siembra, floración y control del cultivo y otros.

REFERENCES

Bosy, J., Rohm, W., Borkowski, A., Kroszczynski, K., & Figurski, M. (2010). Integration and verification of meteorological observations and NWP model data for the local GNSS tomography. Atmospheric Research, 96(4), 522–530. https://doi.org/10.1016/j.atmosres.2009.12.012

Chen, G., Li, S., Knibbs, L. D., Hamm, N. A. S., Cao, W., Li, T., Guo, J., Ren, H., Abramson, M. J., & Guo, Y. (2018). A machine learning method to estimate PM2.5 concentrations across China with remote sensing, meteorological and land use information. Science of the Total Environment, 636, 52–60. https://doi.org/10.1016/j.scitotenv.2018.04.251

Colston, J. M., Ahmed, T., Mahopo, C., Kang, G., Kosek, M., Junior, F. de S., Shrestha, P. S., Svensen, E., Turab, A., Zaitchik, B., & Network, T. M.-E. (2018). Evaluating meteorological data from weather stations, and from satellites and global models for a multi-site epidemiological study. Environmental Research, 165, 91–109.

Friedman, J. H. (2001). A gradient boosting machine. IMS 1999 Reitz Lecture, №3, 39.

Gonçalves, A. M., Silva, J. G., & Gomes, P. M. V. (2006). Meteorological support to forest fire prevention. In Forest Ecology and Management (Vol. 234, p. S41). https://doi.org/10.1016/j.foreco.2006.08.062

Han, H., Lee, S., Im, J., Kim, M., Lee, M. I., Ahn, M. H., & Chung, S. R. (2015). Detection of convective initiation using Meteorological Imager onboard Communication, Ocean, and Meteorological Satellite based on machine learning approaches. Remote Sensing, 7(7), 9184–9204. https://doi.org/10.3390/rs70709184

IBM's AutoAI at work: two real-world applications | by Álvaro Corrales Cano | IBM Garage | Medium. (n.d.).

Kok, M., Smith, J. G., Wohl, C. J., Siochi, E. J., & Young, T. M. (2015). Critical considerations in the mitigation of insect residue contamination on aircraft surfaces - A review. In Progress in Aerospace Sciences (Vol. 75, pp. 1–14). https://doi.org/10.1016/j.paerosci.2015.02.001

Lu, H., Wu, Y., Li, Y., & Liu, Y. (2017). Effects of meteorological droughts on agricultural water resources in southern China. In Journal of Hydrology (Vol. 548, pp. 419–435). https://doi.org/10.1016/j.jhydrol.2017.03.021

Ma, P., Wang, S., Zhou, J., Li, T., Fan, X., Fan, J., & Wang, S. (2020). Meteorological rhythms of respiratory and circulatory diseases revealed by Harmonic Analysis. In Heliyon (Vol. 6, Issue 5). https://doi.org/10.1016/j.heliyon.2020.e04034

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. Frontiers in Neurorobotics, 7(DEC). https://doi.org/10.3389/fnbot.2013.00021

Riabani Mercado, F., García Fernández, W., & Herrera Acebey, J. A. (2016). Sistema de inteligencia artificial para la predicción temprana de heladas meteorológicas. Acta Nova, 7(4), 483–495.

Rozenstein, O., & Karnieli, A. (2011). Comparison of methods for land-use classification incorporating remote sensing and GIS inputs. In Applied Geography (Vol. 31, Issue 2, pp. 533–544). https://doi.org/10.1016/j.apgeog.2010.11.006

Sotelo, S., Guevara, E., Llanos-Herrera, L., Agudelo, D., Esquivel, A., Rodriguez, J., Ordoñez, L., Mesa, J., Muñoz Borja, L. A., Howland, F., Amariles, S., Rojas, A., Valencia, J. J., Segura, C. C., Grajales, F., Hernández, F., Cote, F., Saavedra, E., Ruiz, F., … Ramirez-Villegas, J. (2020). Pronosticos AClimateColombia: A system for the provision of information for climate risk reduction in Colombia. Computers and Electronics in Agriculture, 174. https://doi.org/10.1016/j.compag.2020.105486

Wardah, T., Abu Bakar, S. H., Bardossy, A., & Maznorizan, M. (2008). Use of geostationary meteorological satellite images in convective rain estimation for flash-flood forecasting. Journal of Hydrology, 356(3–4), 283–298. https://doi.org/10.1016/j.jhydrol.2008.04.015