Modelo de detección de niebla en ruta mediante aplicación de modelos estadísticos y de minería de datos con herramienta de software libre

Marcos, Carlos Eduardo; Martínez Micakoski, Fernanda; Perez Angueira, Luciana; Gomez, Jonathan; Perez Angueira, Angeles; Marcos, Candela; Molina, Christian; Blasco, Lucas; Benuzzi, Germán
Facultad Regional Trenque Lauquen, Universidad Tecnológica Nacional.
Racedo 298. Trenque Lauquen, Bs. As., Argentina.
marcoscarloseduardo@gmail.com

Resumen

El tema abordado es la identificación de variables que permitan predecir la presencia de niebla en rutas con la finalidad de limitar la adquisición de datos a los momentos en que se dan tales condiciones en el tramo de la Ruta Nacional 33 en inmediaciones de la ciudad de Trenque Lauquen, Provincia de Buenos Aires, Argentina. El objetivo final es el desarrollo de un prototipo de captura de datos remoto equipado con sensores y cámara de video que permita caracterizar la zona en presencia de niebla donde se realizan intervenciones en la cinta asfáltica para mejorar la seguridad vial. Los datos históricos fueron provistos por el Servicio Meteorológico Nacional. Se realizaron dos modelos de predicción de la variable "niebla", uno basado en la técnica estadística de Modelos Lineales Generalizados y otro mediante Árboles de Clasificación con bagging. Ambos modelos seleccionaron las mismas variables (hora local, velocidad del viento, temperatura y humedad relativa) para predecir la presencia de niebla a partir de los datos históricos con una exactitud del 80% sobre los datos de testeo.

1. Introducción

En el marco del proyecto de investigación "Valoración del desempeño de modelos de soluciones viales a nivel de calzada para la conducción segura bajo condición de escasa visibilidad por niebla", homologado por la Secretaria de Ciencia, Tecnología y Posgrado de la Universidad Tecnológica Nacional es necesario caracterizar la zona bajo análisis en presencia de niebla y obtener datos que permitan evaluar la eficacia de la intervención realizada en la cinta asfáltica. Para ello se desarrollará un prototipo de adquisición de datos de bajo costo, basado en la Arduino [3], con sensores y una cámara de video.

Dado que los sitios de aplicación se ubican sobre las rutas y no poseen cobertura para la transmisión ni capacidad de almacenamiento de un gran volumen de datos es necesario reducir la adquisición de los mismos a los momentos en los que existe una mayor posibilidad de presencia de niebla. Por este motivo se solicitó al Servicio Meteorológico Nacional (SMN) el registro histórico de la estación local y las dos más próximas a la ciudad de Trenque Lauquen.

Mediante el análisis estadístico de estos registros se desea identificar las variables que mejor predicen la presencia de niebla de manera que el prototipo inicie la adquisición de los datos en el tramo bajo estudio solo bajo esta condición.

La Minería de Datos o Explotación de Información, es el proceso de extraer conocimiento útil, comprensible y novedoso de grandes volúmenes de datos, siendo su principal objetivo encontrar información oculta o implícita, que no es posible obtener mediante métodos estadísticos convencionales. La entrada al proceso de minería está formada por contenedores de información diversos, esto incluye bases de datos relacionales, almacenes de datos (Datawarehouse), documentos en texto libre, datos de la Web, entre otros [14].

Aunque hace mucho que existen algunas de las técnicas de procesamiento de datos, antes solo podían permitírselas los organismos de seguridad del estado, los laboratorios de investigación y las mayores compañías del mundo. Ahora muchas de estas herramientas se han democratizado (aunque no así los datos) [19].

La técnica SEMMA, acrónimo de las cinco fases del proceso: Sample (Muestreo), Explore (Exploración), Modify (Modificación), Model (Modelado), Assess (Evaluación), surge en el año 2000 y tiene como fin establecer las etapas principales de un proceso de minería de datos [24].

Dentro de los modelos predictivos se buscan aquellos que permitan obtener una fácil interpretación y ecuaciones

matemáticas o reglas que se puedan programar en el prototipo ya que el proceso de predicción debe tener lugar en el sitio remoto. Adicionalmente se requieren modelos cuya variable respuesta tenga una distribución binomial, como es el caso de la presencia o no de niebla.

Nelder y Wedderburn estudiaron los Modelos Lineales Generalizados (glm) incorporando de esta manera la posibilidad de modelar variables respuestas continuas o categóricas con distribuciones del error no necesariamente homocedásticas [20]

Los árboles de clasificación y regresión (CART) fueron desarrollados en los años 80 por Breiman, Freidman, Olshen y Stone [7].

La metodología CART utiliza datos históricos para construir árboles de clasificación o de regresión los cuales son usados para clasificar o predecir nuevos datos. Estos árboles CART pueden manipular fácilmente variables numéricas y/o categóricas. Entre otras ventajas está su robustez a outliers, la invarianza en la estructura de sus árboles de clasificación o de regresión a transformaciones monótonas de las variables independientes, y sobre todo, su interpretabilidad [10].

El objetivo principal de este trabajo es obtener las variables predictoras de presencia de niebla en la ruta en base a los datos meteorológicos históricos aplicando y evaluando comparativamente el uso de una técnica estadística (glm) y otra de minería de datos (árboles de clasificación) sobre datos públicos y mediante el uso de software libre [4].

2. Metodología

2.1. Variables

El proceso de análisis se basó en la técnica SEMMA.

Para realizar este trabajo se solicitó al Servicio Meteorológico Nacional los datos históricos de las estaciones Trenque Lauquen (87540), Pehuajó Aero (87544) y Pigüé Aero (87679).

El archivo con las observaciones se importó para ser procesado mediante el software libre R [22].

Las variables disponibles son las siguientes (ver Tabla 1):

Tabla 1. Variables de estudio.

Variable	Tipo	Posibles Valores
Estación	Cualitativa	Pigüé, Pehuajó, Trenque Lauquen
Fecha	Cronológica	Fechas válidas
Hora Local	Cuantitativa	Enteros de 0 a 23
Temperatura	Cuantitativa	Racionales
Humedad Relativa	Cuantitativa	Enteros positivos
Dirección Viento	Cuantitativa	Enteros de 0 a 36 (decagrados)
Velocidad Viento	Cuantitativa	Enteros positivos incluyendo el 0
Visibilidad	Cuantitativa	Expresiones alfanuméricas codificadas
Código Tiempo Presente	Cualitativa	Enteros de 0 a 99

Durante la exploración de las variables se determinó que la vista minable posee 5.969 observaciones. La primera es del 01 de Marzo de 2001 y la última del 13 de Agosto de 2016. De los 100 códigos de tiempo que utiliza el SMN, 17 están asociados a fenómenos con niebla o neblina.

La variable a predecir es *niebla*, por lo que se agregó dicha variable a la tabla minable a partir de los códigos de tiempo presente que hacen referencia a *niebla* y con una *visibilidad* máxima de hasta 1 km para diferenciarla del fenómeno *neblina*.

Se reemplazó *fecha* por las tres variables *día*, *mes* y *año*, generadas a partir de la misma.

Para poder modelizar se determinó que 32 observaciones poseían datos faltantes y dado que la proporción era mínima en función del total de la tabla se procedió a eliminarlas.

La proporción de la variable *respuesta* (niebla) positiva era del 30% de las observaciones.

Para verificar la homogeneidad de las muestras entre las tres estaciones primero se identificó si el mismo debía ser paramétrico o no paramétrico y luego se testeó si se podía asumir que todas provenían de una misma población.

Como cada estación poseía más de 50 observaciones se empleó la prueba de Kolmogorov-Smirnov con la corrección de Lilliefors [17] verificando que la mayoría de las variables no sigue una distribución normal.

Se realizó el test de Levene [16] para determinar si podíamos asumir que todas las observaciones provenían de una misma población, con resultando negativo, por lo que es necesario plantear un modelo para cada estación. En particular se continuó con la estación Trenque Lauquen debido a las necesidades del proyecto de investigación.

Visualmente se procedió a conocer la distribución de las variables. Como ejemplo se puede observar en la distribución de la variable humedad relativa (ver Figura 1), en un gráfico focalizado en valores superiores al 80%. En gris se representan las observaciones donde no se observó presencia de niebla y en naranja en los que sí. Por lo tanto es posible encontrar niebla con valores del 91 al 100% de humedad relativa, pero la proporción de casos positivos supera a los negativos a partir del 96%.

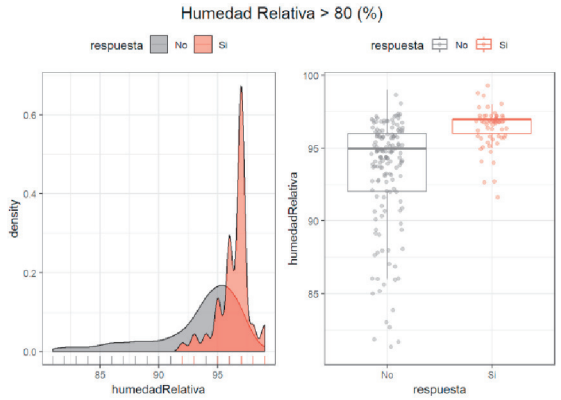

Figura 1. Distribución variable humedad relativa.

Se pudo determinar que las observaciones entre las 22 y las 7 horas son prácticamente inexistentes por lo que el modelo no será confiable para predecir la presencia de niebla en ese rango horario (ver Figura 2).

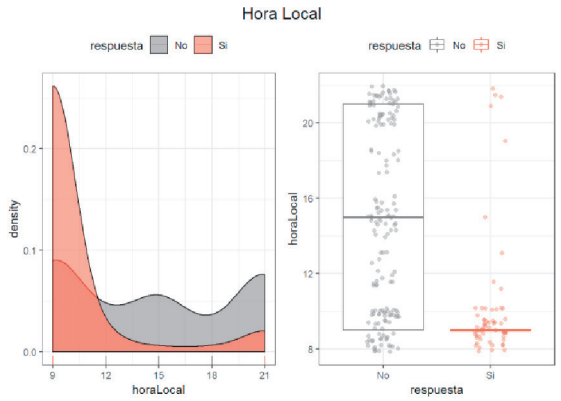

Figura 2. Distribución variable hora local.

Las variables *día* y *año* poseían una distribución aleatoria respecto de la presencia de niebla por lo que se las descartó de las posibles variables predictoras para el modelo.

Se realizó un Análisis de Componentes Principales (PCA) [15] y se pudo establecer que la correlación entre las variables es débil, con excepción de una mediana correlación positiva entre la *dirección* y la *velocidad del viento*, y una débil a mediana correlación negativa entre la *humedad relativa* y la *velocidad del viento*.

Se comparó la distribución de las observaciones en base a las dos dimensiones principales con una distribución simulada aleatoria para verificar si realmente es posible detectar algún tipo de agrupamiento (ver Figura 3).

Figura 3. Distribución variable humedad relativa.

Se detectó la existencia de al menos dos clusters y se identificaron potenciales valores atípicos. Se calculó el estadístico Hopkins [5] para estudiar la distribución espacial obteniendo un valor de 0.28 confirmando que existe algún tipo de agrupamiento. Mediante los métodos Elbow, Average Silhouette y Gap Statistics [25] se determinó el número óptimo de clusters en 2.

A través de la función *createDataPartition* del paquete Caret en R se realizó una partición equilibrada de los datos mediante un muestreo aleatorio dentro de cada clase de la variable respuesta, preservando la distribución general de la clase de los datos. Se asignaron el 70% de los mismos para entrenamiento y el 30% restante para testeo de los modelos.

2.2. Modelo Lineal Generalizado

En primer lugar se generó un Modelo Lineal Generalizado el cual contempla la restricción $0 \leq (E(Y_i|X_i) \leq 1$ donde *0* representa la categoría "No" y *1* la categoría "Sí" de la variable respuesta *niebla*.

Dentro de las funciones monótonas crecientes, en forma de S, se optó por la función logística, porque:

- Desde el punto de vista matemático, es una función extremadamente flexible y fácil de utilizar.
- Tiene una interpretación relativamente sencilla.
- La evidencia empírica ha demostrado que este modelo es adecuado en la mayoría de los casos en los cuales la respuesta es binaria.

El modelo logístico tiene la forma:

$$E(y) = \frac{e^{x'\beta}}{1 + e^{x'\beta}}$$

Donde x es el vector de variables explicatorias y β es el vector de parámetros. Esta ecuación también puede expresarse como:

$$E(y) = \frac{1}{1 + e^{-x'\beta}}$$

o sea:

$$\pi_i = \frac{1}{1 + e^{-x'\beta}}$$

que es equivalente a:

$$1 - \pi_i = \frac{1}{1 + e^{x'\beta}}$$

Con lo cual se tiene que:

$$\frac{\pi_i}{1 - \pi_i} = \frac{1 + e^{x'\beta}}{1 + e^{-x'\beta}} = e^{x'\beta}$$

A esta transformación se la conoce como transformación *logit* de la probabilidad π_i y la relación $\frac{\pi_i}{1-\pi_i}$ una razón de probabilidades o ventaja (*odds ratio*).
Si se toma el logaritmo natural, se obtiene

$$Ln\left(\frac{\pi_i}{1 - \pi_i}\right) = x'\beta$$

Con lo cual se tiene que el logaritmo de la razón de probabilidades es lineal, tanto en las variables como en los parámetros. La estimación de estos puede realizarse mediante el método de máxima verosimilitud [13].
La forma general del modelo *logit* se puede expresar como:

$$y_i = E(y_i) + \varepsilon_i$$

donde las observaciones y_i son variables aleatorias independientes Bernoulli, con valores esperados:

$$E(y_i) = \pi_i = \frac{e^{x'\beta}}{1 + e^{x'\beta}}$$

Como cada observación sigue una distribución Bernoulli, su distribución será:

$$f_i(y_i) = \pi_i^{y_i}(1 - \pi_i)^{1-y_i}, i = 1, 2, 3, \dots, n$$

Y dado que las observaciones son independientes, la función de verosimilitud será:

$$L(y_1, y_2, \dots, y_n, \beta) = \prod_{i=1}^{n} f_i(y_i) = \prod_{i=1}^{n} \pi_i^{y_i}(1 - \pi_i)^{1-y_i}$$

Al tomar logaritmo a la función de verosimilitud:

$$LnL(y_1, y_2, \dots, y_n, \beta) = Ln \prod_{i=1}^{n} f_i(y_i)$$

$$= \sum_{i=1}^{n} \left[y_i Ln\left(\frac{\pi_i}{1 - \pi_i}\right)\right] + \sum_{i=1}^{n} Ln(1 - \pi_i)$$

Como

$$1 - \pi_i = \frac{1}{1 + e^{x'\beta}} \quad \text{y} \quad Ln\left(\frac{\pi_i}{1 - \pi_i}\right) = x'\beta,$$

El logaritmo de la verosimilitud se puede expresar para el modelo de regresión logística:

$$Ln\,(y, \beta) = \sum_{i=1}^{n} y_i x_i'\beta - \sum_{i=1}^{n} Ln\left[1 + e^{x_i'\beta}\right]$$

Los estimadores de máxima verosimilitud se pueden obtener mediante un algoritmo de mínimos cuadrados iterativamente re ponderados.
Si $\hat{\beta}$ es el estimador obtenido, mediante el método iterativo y siendo ciertas las hipótesis del modelo, se puede demostrar que en forma asintótica:

$$E(\hat{\beta}) = \beta \quad \text{y} \quad V(\hat{\beta}) = (X'V^{-1}X)^{-1}$$

El valor estimado del predictor lineal es $\hat{\eta}_i = x_i\hat{\beta}$, y el valor esperado del modelo de regresión logístico, se puede expresar:

$$\hat{y}_i = \hat{\pi}_i = \frac{e^{\hat{\eta}_i}}{1 + e^{\hat{\eta}_i}} = \frac{e^{(x_i'\hat{\beta})}}{1 + e^{(x_i'\hat{\beta})}} = \frac{1}{1 + e^{(-x_i'\hat{\beta})}}$$

Para el cálculo de los parámetros se utilizó la función *glm* del software R, con una distribución de la variable dependiente binominal y la función de transformación *logit* obteniéndose el primer modelo *glm1*.

Para mejorar la performance de este modelo inicialmente se identificaron valores atípicos a través del gráfico de residuos estandarizados y del test de Bonferonni [9], el cual ajusta el nivel de significación en relación al número de pruebas estadísticas realizadas simultáneamente sobre un conjunto de datos. El nivel de significación para cada prueba se calcula dividiendo el error global de tipo I entre el número de pruebas a realizar. El ajuste de Bonferroni se considera conservador. Eliminados estos valores atípicos se obtuvo el modelo *glm2*.

La comparación entre los distintos modelos se realizó en base al Criterio de Información de Akaike (AIC) [1], que es una herramienta objetiva que permite cuantificar la idoneidad de un modelo particular en relación a un conjunto finito de modelos [A].

Este criterio, que se enmarca en el campo de la teoría de la información, se define como:

$$AIC \; = \; -2\ln\left(L(\hat{\theta})\right) + \; 2K$$

donde $\ln\left(L(\hat{\theta})\right)$ es el logaritmo de la máxima verosimilitud, que permite determinar los valores de los parámetros libres de un modelo estadístico [2], y K es el número de parámetros libres del modelo. Esta expresión proporciona una estimación de la distancia entre el modelo y el mecanismo que realmente genera los datos observados, que es desconocido y en algunos casos imposibles de caracterizar. Como la estimación se hace en función de los datos experimentales, esta distancia es siempre relativa y dependiente del conjunto de datos experimentales. Por tanto, un valor individual de AIC no es interpretable por sí solo, y los valores AIC sólo tienen

sentido cuando se realizan comparaciones utilizando los mismos datos experimentales [18].

Se evaluó la significancia de los parámetros del modelo *glm2* y sus intervalos de confianza, procediendo a generar un nuevo modelo *glm3* con foco en la eliminación de las variables menos significativas a través de un proceso de eliminación hacia atrás (Backward Stepwise Regression). Se partió con todas las variables seleccionadas del modelo *glm2* y mediante iteración se generaron todos los modelos que se pueden crear eliminando un único predictor a la vez, optando por el modelo con menor AIC.

Por último se evaluó agregar al modelo nuevas variables a partir de la interacción de los predictores existentes.

Para los distintos modelos se calculó la tasa de acierto para los datos de entrenamiento y testeo, como así también el área bajo la curva ROC (Receive Operating Characteristics) [12]. Un gráfico ROC muestra el rendimiento de un método de clasificación binaria con salida ordinal continua o discreta. Muestra la sensibilidad (la proporción de observaciones positivas correctamente clasificadas) y la especificidad (la proporción de observaciones negativas correctamente clasificadas) a medida que el umbral de salida se mueve sobre el rango de todos los valores posibles. Las curvas ROC no dependen de las probabilidades de clase, lo que facilita su interpretación y comparación entre diferentes conjuntos de datos [23].

2.3. Árboles de Decisión

Un árbol de decisión es un modelo de predicción cuyo objetivo principal es el aprendizaje inductivo a partir de observaciones y construcciones lógicas. Son muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva para la solución de un problema. El conocimiento obtenido durante el proceso de aprendizaje inductivo se representa mediante un árbol. En el campo del aprendizaje automático, hay distintas maneras de obtener árboles de decisión, la que utilizamos es una técnica de aprendizaje supervisado conocida como CART: Classification And Regression Trees. El objetivo es obtener una función que nos permita predecir, a partir de variables independientes, el valor de la variable dependiente para casos desconocidos. Dado que la variable a predecir es discreta, se utilizó un árbol de clasificación.

La implementación particular de CART utilizada es conocida como Recursive Partitioning and Regression Trees o RPART.

De manera general, lo que hace este algoritmo es encontrar la variable independiente que mejor separa nuestros datos en grupos, que corresponden con las categorías de la variable objetivo. Esta mejor separación es expresada con una regla. A cada regla corresponde un nodo. Una vez hecho esto, los datos son separados (particionados) en grupos a partir de la regla obtenida. Después, para cada uno de los grupos resultantes, se repite el mismo proceso. Se busca la variable que mejor separa los datos en grupos, se obtiene una regla, y se separan los datos. Se realiza esto de manera recursiva hasta que es imposible obtener una mejor separación. Cuando esto ocurre, el algoritmo se detiene. Cuando un grupo no puede ser partido mejor, se le llama nodo terminal u hoja.

Una característica muy importante en este algoritmo es que una vez que alguna variable ha sido elegida para separar los datos, ya no es usada de nuevo en los grupos que ha creado. Se buscan variables distintas que mejoren dicha separación.

Además, supongamos que después de una partición que ha creado dos grupos, A y B. Es posible que para el grupo A, la variable que mejor separa estos datos sea diferente a la que mejor separa los datos en el grupo B. Una vez que los grupos se han separado, al algoritmo "no ve" lo que ocurre entre grupos, estos son independientes entre sí y las reglas que aplican para ellos no afectan en nada a los demás.

El proceso de construcción de un árbol de predicción se divide en dos etapas:

- División sucesiva del espacio de los predictores generando regiones no solapantes (nodos terminales) $R_1, R_2, R_3, \ldots, R_j$.
- Predicción de la variable respuesta en cada región.

A pesar de la sencillez con la que se puede resumir el proceso de construcción de un árbol, es necesario establecer una metodología que permita crear las regiones $R_1, R_2, R_3, \ldots, R_j$, o lo que es equivalente, decidir donde se introducen las divisiones: en que predictores y en que valores de los mismos.

Para poder determinar cuáles de todas las divisiones son las óptimas se utilizó el índice de Gini, el cual es una medida de la varianza total en el conjunto de las K clases del nodo m. Se considera una medida de pureza del nodo.

$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

donde $\hat{p}_{mk}$representa la proporción de observaciones del nodo m que pertenecen a la clase k. Cuando $\hat{p}_{mk}$ es cercano a 0 o a 1 (el nodo contiene mayoritariamente observaciones de una clase), el término $\hat{p}_{mk}(1 - \hat{p}_{mk})$ es muy pequeño. Como consecuencia, cuanto mayor sea la pureza del nodo, menor el valor del índice Gini G [21].

Para cada posible división se calcula el valor del índice en cada uno de los dos nodos resultantes. Se suman los dos valores ponderando cada uno por la fracción de observaciones que contiene cada nodo.

$$\frac{n_A}{n_{Total}} \times pureza\ A + \frac{n_B}{n_{Total}} \times pureza\ B$$

La división con menor valor se selecciona como división óptima.

Tras la creación de un árbol, las observaciones de entrenamiento quedan agrupadas en los nodos terminales. Para predecir una nueva observación, se recorre el árbol en función del valor de sus predictores hasta llegar a uno de los nodos terminales.

Las principales ventajas del método de clasificación son su interpretabilidad, pues nos da un conjunto de reglas a partir de las cuales se pueden tomar decisiones. Este es un algoritmo que no es demandante en poder de cómputo comparado con procedimientos más sofisticados y, a pesar de ello, tiende a dar buenos resultados de predicción para muchos tipos de datos.

Sus principales desventajas son que este es un tipo de clasificación "débil", pues sus resultados pueden variar mucho dependiendo de la muestra de datos usados para entrenar un modelo. Además es fácil sobre ajustar los modelos, esto es, hacerlos excelentes para clasificar datos que conocemos, pero deficientes para datos conocidos.

Al igual que todo modelo estadístico, los árboles de predicción sufren el problema del equilibrio *bias-varianza*. El término *bias* hace referencia a cuánto se alejan en promedio las predicciones de un modelo respecto a los valores reales, es decir, cómo se aproxima el modelo a la relación real entre las variables. La *varianza* hace referencia a cuánto varía el modelo dependiendo de la muestra empleada en el entrenamiento. A medida que se aumenta la complejidad de un modelo, se dispone de mayor flexibilidad para adaptarlo a las observaciones, reduciendo así el *bias* y mejorando su capacidad predictiva. Sin embargo, alcanzado un determinado grado de flexibilidad, aparece el problema de *overfitting*, el modelo se ajusta tanto a los datos de entrenamiento que es incapaz de predecir correctamente nuevas observaciones. El mejor modelo es aquel que consigue un equilibro óptimo entre *bias* y *varianza*.

Para intentar mejorar el modelo se utilizó la técnica de *bagging* [6]. El término *bagging*, diminutivo de *bootstrap aggregation*, hace referencia al empleo del muestreo repetido (*bootstrapping*) con el fin de reducir la varianza de algunos métodos de aprendizaje estadístico, entre ellos los árboles de predicción.

Dadas n muestras de observaciones independientes Z_1, ..., Z_1, cada una con varianza σ^2, la varianza de la media de las observaciones $\bar{Z}$ es σ^2/n. En resumen, promediando un conjunto de observaciones se reduce la varianza. Basándose en esta idea, una forma de reducir la varianza y aumentar la precisión de un método predictivo es obtener múltiples muestras de la población, ajustar un modelo distinto con cada una de ellas, y hacer la moda de las predicciones resultantes. Si bien en la práctica no se suele tener acceso a múltiples muestras, se puede simular el proceso recurriendo a *bootstrapping*, generando así pseudo-muestras con los que ajustar diferentes modelos y después agregarlos. A este proceso se le conoce como *bagging* y es aplicable a una gran variedad de métodos de regresión. En el caso particular de los árboles de decisión, *bagging* ha demostrado incrementar en gran medida la precisión de las predicciones. La forma de aplicarlo es:

- Generar *B pseudo-training sets* mediante *bootstrapping* a partir de la muestra de entrenamiento original.
- Entrenar un árbol con cada una de las *B* muestras del paso 1. Cada árbol se crea sin apenas restricciones y no se somete a *pruning*, por lo que tiene *varianza* alta pero poco *bias*. En la mayoría de casos, la única regla de parada es el número mínimo de observaciones que deben tener los nodos terminales. El valor óptimo de este hiperparámetro puede obtenerse comparando el *out of bag error* que puede interpretarse de la misma forma que el error de validación cruzada [26].
- Para cada nueva observación, obtener la predicción de cada uno de los *B* árboles. El valor final de la predicción se obtiene como la clase predicha más frecuente (moda) para las variables cualitativas.

Finalmente también se calculó el área bajo la curva ROC y se compararon los resultados con los modelos *glm*.

3. Resultados obtenidos

3.1. Modelo lineal generalizado

3.1.1. Modelo 1 (glm1). Se procedió a generar un modelo estadístico basado en el Modelo Lineal Generalizado biomial por tratarse de una variable respuesta con dos estados posible "Sí" y "No" a la presencia de niebla.

La proporción de la varianza que explica el primer modelo fue 64.55 % sobre los datos de entrenamiento.

Excepto *mes*, el resto de los coeficientes son significativos ($p < 0.05$). La desviación del modelo nulo es de 181.7 con 155 grados de libertad y la de los residuos de 64.4 con 149 grados de libertad. El Criterio de Información de Akaike (AIC) es 78.4 (ver Figura 4).

A través del gráfico de residuos estandarizados frente a valores predichos se identificaron 3 observaciones atípicas, de las cuáles una vez analizadas se determinó que solo una sería retirada del grupo de entrenamiento por tratarse realmente de un valor atípico. El test de Bonferonni para valores atípicos confirmó esta decisión.

```
Call:
glm(formula = respuesta ~ ., family = "binomial", data = datos_train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.10883  -0.19128  -0.03501   0.04370   2.90384

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)             -101.26673   29.71863  -3.408 0.000656 ***
mes                       -0.05762    0.17285  -0.333 0.738847
horaLocal                 -0.36314    0.11365  -3.195 0.001397 **
temperaturaGradosCelsius  -0.40117    0.12051  -3.329 0.000872 ***
humedadRelativa            1.17811    0.32881   3.583 0.000340 ***
direccionVientoDecagrados -0.09601    0.04076  -2.355 0.018506 *
velocidadVientoKmh        -0.37822    0.11316  -3.342 0.000830 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 181.738  on 155  degrees of freedom
Residual deviance:  64.422  on 149  degrees of freedom
AIC: 78.422

Number of Fisher Scoring iterations: 9
```

Figura 4. Modelo glm1

3.1.2. Modelo 2 (glm2). Se procedió a generar un nuevo modelo sin este valor atípico durante el entrenamiento el cual arrojó una proporción de varianza explicativa del 69.89% y un AIC de 67.9.

El intervalo de confianza de los predictores se describe en la Tabla 2:

Tabla 2. Intervalo de confianza predictores modelo glm2.

	2.5 %	97.5 %
(Intercept)	-191.60	-51.62
mes	-0.46	0.29
horaLocal	-1.05	-0.29
temperaturaGradosCelsius	-0.77	-0.21
humedadRelativa	0.66	2.23
direccionVientoDecagrados	-0.22	-0.03
velocidadVientoKmh	-0.78	-0.23

Se observa que el intervalo de confianza para el coeficiente de *mes* contiene el cero, lo que confirma que podría no ser significativo.

La exactitud del modelo *glm2* con datos de entrenamiento fue de 0.909 (ver Tabla 3) y con datos de testeo fue de 0.803 (ver Tabla 4).

Tabla 3. Matriz de confusión del modelo glm2 con datos de entrenamiento.

Predicción	Real	
	No	Si
No	108	8
Si	6	33

Tabla 4. Matriz de confusión del modelo glm2 con datos de testeo.

Predicción	Real	
	No	Si
No	42	7
Si	6	11

Los valores medios de las curvas ROC para los datos de entrenamiento y testo fueron 0.974 y 0.880 respectivamente (ver Figura 5).

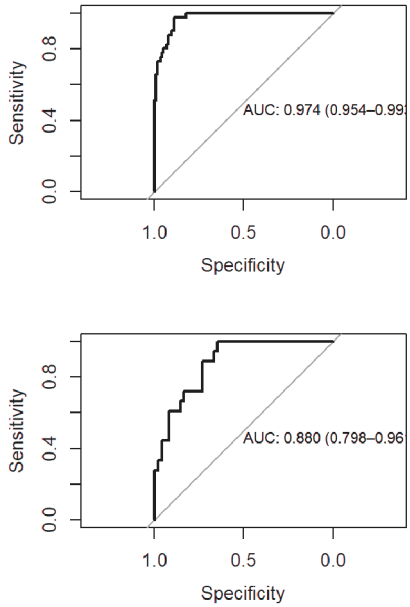

Figura 5. Curvas ROC para modelo glm2

3.1.3. Modelo 3 (glm3). Antes de descartar la variable numérica *mes* se optó por transformarla en categórica pero el modelo generado mantuvo su condición de no significativa e incrementó el AIC a 71.4. Por lo tanto se procedió a ajustar el modelo encontrándose que el mismo únicamente mejora eliminando la variable *mes* (ver Figura 6). El valor inicial de AIC de 67.92 se reduce a 66.07.

```
Start:  AIC=67.92
respuesta ~ mes + horaLocal + temperaturaGradosCelsius + humedadRelativa +
    direccionVientoDecagrados + velocidadVientoKmh

                           Df Deviance    AIC
- mes                       1   54.065 66.065
<none>                          53.917 67.917
- direccionVientoDecagrados  1   61.460 73.460
- temperaturaGradosCelsius   1   72.374 84.374
- velocidadVientoKmh         1   75.037 87.037
- humedadRelativa            1   78.606 90.606
- horaLocal                  1   79.711 91.711

Step:  AIC=66.07
respuesta ~ horaLocal + temperaturaGradosCelsius + humedadRelativa +
    direccionVientoDecagrados + velocidadVientoKmh

                           Df Deviance    AIC
<none>                          54.065 66.065
- direccionVientoDecagrados  1   61.509 71.509
- temperaturaGradosCelsius   1   74.391 84.391
- velocidadVientoKmh         1   75.519 85.519
- humedadRelativa            1   78.917 88.917
- horaLocal                  1   79.818 89.818
```

Figura 6. Modelo glm3

La exactitud del modelo *glm3* reducido se mantuvo y las matrices confusión obtenidas son idénticas a las del modelo *glm2*, confirmando que la variable *mes* no era significativa.

No se observó ninguna mejora intentando agregar combinación de los predictores del modelo reducido (ver Figura 7).

El estadístico de Durbin-Watson [11] arrojó un valor de 1.846 con p-value = 0.145 indicando que no existen problemas de autocorrelación entre los residuos del modelo por lo que no tuvimos motivos para descartarlo.

```
Start:  AIC=66.07
respuesta ~ horaLocal + temperaturaGradosCelsius + humedadRelativa +
    direccionVientoDecagrados + velocidadVientoKmh

                                                    Df Deviance    AIC
<none>                                                   54.065 66.065
+ direccionVientoDecagrados:velocidadVientoKmh       1   52.236 66.236
+ temperaturaGradosCelsius:velocidadVientoKmh        1   52.549 66.549
+ temperaturaGradosCelsius:direccionVientoDecagrados 1   52.853 66.853
+ horaLocal:direccionVientoDecagrados                1   53.402 67.402
+ temperaturaGradosCelsius:humedadRelativa           1   53.740 67.740
+ horaLocal:velocidadVientoKmh                       1   53.812 67.812
+ horaLocal:temperaturaGradosCelsius                 1   53.816 67.816
+ horaLocal:humedadRelativa                          1   53.819 67.819
+ humedadRelativa:direccionVientoDecagrados          1   53.823 67.823
+ humedadRelativa:velocidadVientoKmh                 1   53.973 67.973
- direccionVientoDecagrados                          1   61.509 71.509
- temperaturaGradosCelsius                           1   74.391 84.391
- velocidadVientoKmh                                 1   75.519 85.519
- humedadRelativa                                    1   78.917 88.917
- horaLocal                                          1   79.818 89.818
```

Figura 7. Índices de modelos mediante combinación de predictores del modelo glm3

Los valores medios de las curvas ROC para los datos de entrenamiento y testo fueron 0.974 y 0.877 respectivamente (ver Figura 5).

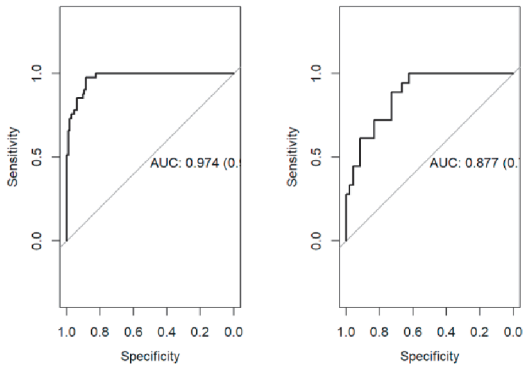

Figura 8. Curvas ROC para modelo glm3

3.2. Árboles de decisión

3.2.1. Árbol de clasificación sin bagging. El modelo obtenido (ver figura 9) se basa en la clasificación mediante la selección de 4 predictores: hora local, velocidad del viento, temperatura y humedad relativa.

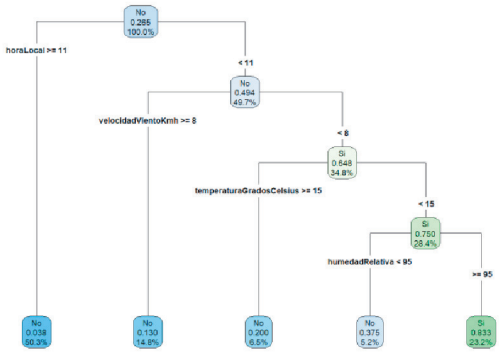

Figura 9. Modelo Árbol de Clasificación

Posee dos ramas principales a partir de la hora local. De las 5 hojas posibles, 1 clasifica positivamente la presencia de niebla (color verde) cuando la hora es inferior a las 11 de la mañana, la velocidad del viento es inferior a 8 km/h, la temperatura no supera los 15 grados y la humedad relativa es igual o superior al 95%.

La importancia de las variables sigue ese mismo orden, desechando el modelo las variables asociadas al mes y a la dirección el viento.

La exactitud del modelo árbol con datos de entrenamiento fue de 0.890 (ver Tabla 6), levemente inferior a la del modelo *glm3*.

Tabla 6. Matriz de confusión del modelo árbol con datos de entrenamiento

Predicción	Real	
	No	Si
No	108	11
Si	6	30

La exactitud del modelo árbol con datos de testeo fue 0.757 (ver tabla 7), un valor inferior al 0.803 del modelo *glm3*.

Tabla 7. Matriz de confusión del modelo árbol con datos de testeo

Predicción	Real	
	No	Si
No	43	11
Si	5	7

3.2.2. Árbol de clasificación con bagging. Para mejorar el modelo se utilizó la técnica de *bagging*. Generamos 100 distintos grupos de entrenamiento mediante bootstrap [4] y se configuró que el número mínimo de observaciones en un nodo sea 3 (2% de los datos de entrenamiento) antes de intentar una división como así también que una división debe pararse cuando el coste factor de complejidad (cp) sea inferior a un 0.001.

La tasa de acierto del modelo con bagging para los datos de entrenamiento fue de 0.994 (ver Tabla 8) y para los datos de testeo de 0.803 (ver Tabla 9).

Tabla 8. Matriz de confusión del modelo árbol usando bagging con datos de entrenamiento.

Predicción	Real	
	No	Si
No	114	1
Si	0	40

Tabla 9. Matriz de confusión del modelo árbol usando bagging con datos de testeo.

Predicción	Real	
	No	Si
No	43	8
Si	5	10

Los valores medios de las curvas ROC para los datos de entrenamiento y testeo fueron 1 y 0.873 respectivamente (ver Figura 10).

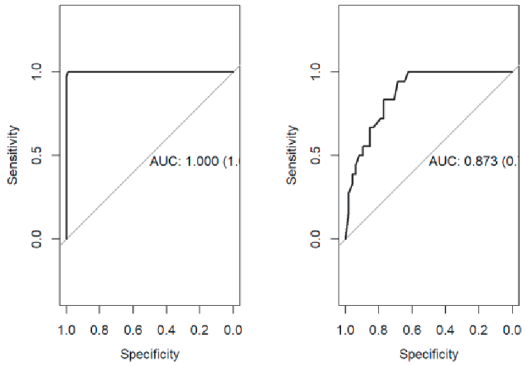

Figura 10. Modelo Árbol de Clasificación con bagging, datos de entrenamiento y testeo respectivamente

A partir del uso de bagging se obtuvo una exactitud del modelo similar a la del modelo *glm3*. El área bajo la curva fue de 0.873 con un intervalo de confianza de 0.790 a 0.955 para los datos de testeo.

El cuadro comparativo de los dos mejores modelos se puede observar a continuación (ver Tabla 10).

Modelo	Tasa Acierto		AUC ROC	
	Entren.	Testeo	Entren.	Testeo
Glm3	0,909	0,803	0,974	0,877
Árbol de clasif. con bagging	0,994	0,803	1,000	0,873

4. Conclusiones

Tanto el modelo *glm3* como el del *árbol de decisión con bagging* seleccionaron las mismas variables para predecir la presencia de niebla a partir de los datos históricos: hora local, velocidad del viento, temperatura y humedad relativa.

La tasa de acierto con ambos modelos para los datos de testeo es de 0.8 por lo que se procederá a programar en el prototipo el modelo *glm3* por su sencillez y menor necesidad de cómputo.

Los valores medios del área bajo la curva ROC fueron de 0.877 para el modelo *glm3* y de 0.873 para el modelo de *árbol de clasificación con bagging*, por lo tanto en

principio podemos considerar que ambos modelos poseen el mismo nivel de discriminación para la variable niebla.

El modelo inicial del árbol de clasificación no tenía un gran poder predictivo en comparación con el Modelo Lineal Generalizado, sin embargo mediante el uso de la técnica de bagging se lo pudo mejorar sustancialmente alcanzando el mismo rendimiento, con el inconveniente de reducir la facilidad de interpretación del modelo.

El software estadístico R permite procesar este volumen de datos para las distintas técnicas estadísticas en tiempos muy razonables en computadoras típicas de uso hogareño.

Los prototipos de captura de datos a desarrollar con estos modelos predictivos serán útiles para reducir la magnitud de almacenamiento de la información en lugares donde no se cuenta con cobertura de banda ancha para la transmisión de los datos.

Esto facilita la realización de un prototipo con sensores que permitan captar estas variables en el sitio bajo estudio y reducir el volumen de datos adquiridos, sobre todo en lo que hace a la captura de video.

Queda pendiente el análisis de las otras estaciones, que de comprobarse que las mismas variables predictoras son elegidas reduciría la complejidad de usar el prototipo en otras ubicaciones a simplemente modificar los parámetros del modelo para las condiciones locales.

5. Reconocimientos

Agradecemos al Servicio Meteorológico Nacional por compartir su base de datos con los registros meteorológicos históricos de la región bajo estudio.

6. Referencias

[1] Akaike, H., "Information theory and an extension of the maximum likelihood principle", *B. N. Petrov y F. Csaki (Eds.). Second International Symposium on Information Theory* (p. 267–281). Budapest: Akademiai Kiado, 1973.

[2] Aldrich, John. "R. A. Fisher and the Making of Maximum Likelihood 1912-1922." *Statistical Science*, vol. 12, no. 3, 1997, pp. 162–176.

[3] Arduino - Introduction. (s. f.). Recuperado 16 de agosto de 2018, de https://www.arduino.cc/en/guide/introduction

[4] Amat Rodrigo, J., "Machine Learning con R y caret" SAS Institute: Data Mining and the Case for Sampling, *https://rpubs.com/Joaquin_AR/383283*, 2018. Recuperado el 18 de agosto de 2018.

[5] Banerjee, A. "Validating clusters using the Hopkins statistic". *IEEE International Conference on Fuzzy Systems, 2004,* pp.149–153.

[6] Breiman, L., "Bagging predictors ", *Machine Learning*, Volume 24, 1996, pp. 123-140.

[7] Breiman L, Friedman J., Olshen R, Stone C. "Classification and regression tres". *Wadsworth & Brooks / Cole Advanced Books and Software*. Monterrey, California, USA, 1984.

[8] Burnham, K. y Anderson, David. "Model Selection and Inference. A practical Information-Theoretic Approach". *Springer Verlag*. New York Inc., USA, 1998.

[9] Cook, R., Weisberg, S., "Residuals and Influence in Regression", New York: *Chapman and Hall*. Recuperado el 18 de Agosto de la University of Minnesota Digital Conservancy, 1982.

[10] Díaz Sepúlveda, J. F., & Correa, J. C. "Comparación entre árboles de regresión CART y regresión lineal". *Comunicaciones en Estadística*, 6(2), 175. https://doi.org/10.15332/s2027-3355.2013.0002.05. 2013.

[11] Durbin J., Watson G., "Testing for Serial Correlation in Least Squares Regression I.", *Biometrika*, 37, 1950, pp.409–428.

[12] Fawcett, T., "An introduction to ROC analysis", *Pattern Recognition Letters*, 27, 2006, pp.861-874.

[13] Green, W. "Análisis Econométrico", *Prentice Hall*, 2011.

[14] Hernández Orallo, Ramirez Quintana y Ferri Ramírez, "Introducción a la Minería de Datos", *Editorial Pearson Prentice Hall*. España, 2004. ISBN 84-205-4091-9.

[15] Jollife, I., "Principal Component Analysis", *Springer*, Second Edition. 2002.

[16] Levene, H., "Robust testes for equality of variances" en Contributions to Probability and Statistics (I. Olkin, ed.) *Stanford Univ. Press*, Palo Alto, CA., 1960, pp.278–292.

[17] Lilliefors, H., "On the Kolmogorov-Smirnov test for normality with mean and variance unknown", *Journal of the American Statistical Association*, 62(318), 399-402, 1967. http://dx.doi.org/10.2307/2283970.

[18] Martínez, D., Albín, J., Cabaleiro, J., Pena, T., Rivera, F. y Blanco, V. "El criterio de información de Akaike en la obtención de modelos estadísticos de Rendimiento". *Conference Paper: XX Jornadas de Paralelismo*, 2009, pp. 439-444.

[19] Mayer-Schönberger, V., Cukier, K., "Big Data. La revolución de los datos masivos", *Turner Noema*, 2013, p.29.

[20] P. McCullagh y J. A. Nelder, "Generalized Linear Models", *Chapman & Hall*, 1992.

[21] Piccarreta, R.: "Ordinal Classification Trees Based on Impurity Measures". *Advances in Multivariate Data Analysis*, *Springer - Verlag*, Berlín, 2004, pp. 39–51.

[22] R Core Team. "R: A language and environment for statistical computing", *R Foundation for Statistical Computing*, Vienna, Austria. URL https://www.R-project.org/. 2017.

[23] Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, JC., y Müller, M. "pROC: an open-source package for R and S+ to analyze and compare ROC curves". *BMC*

Bioinformatics, 12, 2011, pp. 77. DOI: 10.1186/1471-2105-12-77.

[24] SAS Institute: Data Mining and the Case for Sampling, *http://nas.uhcl.edu/boetticher/ML_DataMining/SAS-SEMMA.pdf*, 1998. Recuperado el 17 de agosto de 2018.

[25] Tibshirani, R., Walther, G. and Hastie, T., "Estimating the number of data clusters via the Gap statistic", *Journal of the Royal Statistical Society* B, 63, 2991, pp.411–423.

[26] Zhi-Hua Zhou, "Ensemble Methods: Foundations and Algorithms", *CRC Press*, 2012