

Desarrollo y uso de herramientas libres para la explotación de datos de los radares meteorológicos del INTA

Yanina Bellini Saibene^{1,3}, Martín Volpaccio³, Santiago Banchemo², Romina Mezher²,

¹ EEA INTA Anguil, Ruta Nacional N° 5, km 580, Anguil (6326), La Pampa, Argentina

² Instituto de Clima y Agua, Castelar, Bs.As., Argentina

³ Universidad Austral (Maestría en Data Mining)

{bellini.yanina, banchemo.santiago, mezher.romina}@inta.gob.ar

Resumen. Los radares meteorológicos son herramientas muy útiles para nuestro país, tanto que en 2011 se lanzó el plan de radarización de Argentina. El INTA cuenta con una red de radares meteorológicos para estudiar fenómenos meteorológicos relacionados con la actividad agropecuaria y generar herramientas que permitan acceder, procesar y analizar los datos de la red. Este trabajo presenta la aplicación de un proceso de minería de datos sobre los datos del radar situado en Anguil, La Pampa, para estudiar el fenómeno de granizo utilizando exclusivamente herramientas de software libre. Dentro de las herramientas utilizadas se presenta un desarrollo realizado por INTA para iniciar el reemplazo del software nativo y propietario de los radares y para implementar los modelos obtenidos con el proceso de minería de datos. El software generado permitió aplicar el proceso CRISP-DM de manera exitosa y se integró de forma sencilla con las otras herramientas libres seleccionadas, demostrando ser multiplataforma. El modelo obtenido se pudo utilizar para la clasificación de nuevos eventos presentando un buen rendimiento.

Keywords: RADAR, Agrometeorología, DataMinig, Granizo

1 Introduction

Los radares meteorológicos son herramientas muy útiles para la identificación, análisis, el monitoreo y evaluación de fenómenos hidro-meteorológicos, la previsión de contingencias y los pronósticos a corto plazo, entre otras aplicaciones [1] [2]. Su importancia para el país queda demostrada con el lanzamiento del proyecto Sistema Nacional de Radares Meteorológicos (SINARAME), anunciado por el ejecutivo nacional el 6 de junio de 2011 [1].

El Instituto Nacional de Tecnología Agropecuaria (INTA) cuenta con tres radares ubicados en las Estaciones Experimentales Agropecuarias (EEA) Paraná, Pergamino y Anguil (Fig. 1) [3] generando una red, cuyo objetivo general se puede resumir en mejorar el conocimiento del ambiente climático en el que se desarrollan las plantas y los animales.

La propuesta de INTA incluye el uso de la red de radares para estudiar fenómenos meteorológicos relacionados con la actividad agropecuaria y generar herramientas que permitan acceder, procesar y analizar los datos de la red. De esta forma los interesados podrán aprovechar los conocimientos generados por INTA y contar con un software específico que les permita descargar la información necesaria, transformarla a un formato estándar y generar nuevos productos.

El propósito de este trabajo es presentar el software desarrollado para aplicar el proceso Cross-Industry Standard Process for Data Mining (CRISP-DM) y su aplicación práctica en el análisis del fenómeno del granizo en el este de La Pampa, sur de Córdoba y oeste de Buenos Aires durante el año 2012, utilizando los datos provistos por el radar situado en la EEA Anguil, La Pampa, Argentina.



Fig. 1. Mapa de la red de radares meteorológicos de INTA

2. Aplicación del proceso CRISP-DM en el análisis del fenómeno del granizo.

Uno de los eventos climáticos de importancia a estudiar es el granizo, el cual consiste en un tipo de precipitación formado por partículas de hielo (conglomerado policristalinos) de diámetro igual o superior a 5mm [4]. El granizo es considerado un riesgo agroclimático y produce importantes pérdidas económicas en cultivos [5] [6] [7]. Más del 85% de los seguros agropecuarios en la Argentina son relativos a coberturas de granizo [8].

Para aplicar técnicas de minería de datos para analizar este fenómeno se utilizó el proceso de CRISP-DM. Este proceso es uno de los más difundidos y consta de seis fases concebidas como un proceso cíclico (Fig. 2) [9] [10]. A continuación se describen las herramientas libres desarrolladas y utilizadas para las fases “Comprender los datos” a “Implementación”.

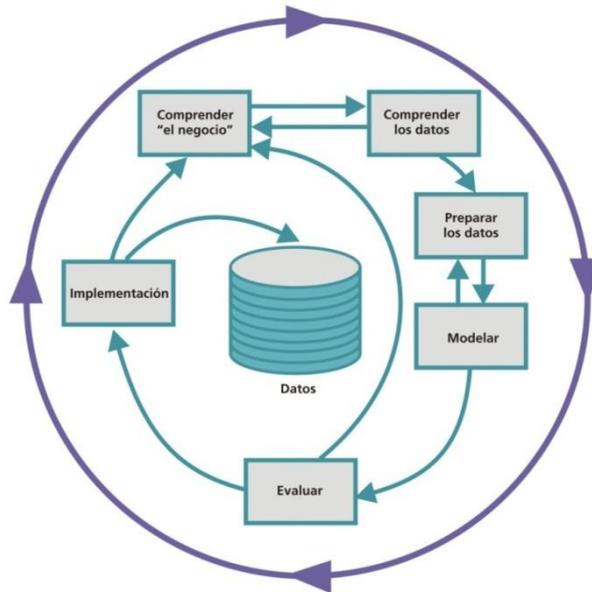


Fig. 2. Fases del proceso de CRISP-DM (Adaptado de: [10])

2.1. Comprender los datos.

Los radares de INTA emiten a la atmósfera pulsos de energía electromagnética en la banda C de la frecuencia de las microondas [3] y reciben la energía reflejada por las partículas presentes en la atmósfera. Esa energía devuelta al radar es medida y se calculan una serie de variables como el factor de reflectividad (Z), la reflectividad diferencial (ZDR), el coeficiente de correlación polarimétrica (Rho_{HV}), el desplazamiento de fase diferencial ($PhiDP$), el desplazamiento de fase diferencial específica (KDP), la velocidad radial (V) y la anchura del espectro (W) [11], las cuales brindan información sobre la forma y tamaño de los hidrometeoros.

Los datos son recolectados por escaneos del volumen que rodea al radar a 120, 240 y 480 kilómetros con giros de la antena de 360 grados en forma horizontal, iniciando con una elevación de 0,5 grados y aumentando el ángulo de elevación 12 veces, hasta llegar a 15,1 grados. Cada escaneo está configurado para ocurrir cada 10 minutos y se almacenan en archivos separados llamados volúmenes [12] [13].

En estos volúmenes contienen el valor de la variable medida para cada unidad de muestreo (1 km^3) en todas las elevaciones y la posición de cada unidad muestreada con respecto del radar en coordenadas polares [13]. Son almacenados en un servidor in situ del radar, al cual solo se puede acceder por vía remota. La Fig. 3 presenta un

listado de volúmenes y el detalle del formato del nombre que lo identifica de forma única.

	Nombre	Fecha de modifica...	Tipo	Tamaño
Fecha: aaaammdd (ej: 20110328, 28 de Marzo de 2011)	2011032803200300KDP.vol	12/07/2013 9:49	Archivo VOL	96 KB
Hora (UTC): hhmmss (ej: 032003, tres y veinte con tres segundos)	2011032803200300RhoHV.vol	12/07/2013 9:49	Archivo VOL	275 KB
Variable (ej: KDP)	2011032803200300ZDR.vol	12/07/2013 9:49	Archivo VOL	181 KB
	2011032803300300dBZ.vol	12/07/2013 9:49	Archivo VOL	176 KB
	2011032803300300KDP.vol	12/07/2013 9:49	Archivo VOL	94 KB
	2011032803300300RhoHV.vol	12/07/2013 9:49	Archivo VOL	278 KB
	2011032803300300ZDR.vol	12/07/2013 9:49	Archivo VOL	181 KB
	2011032803400300dBZ.vol	12/07/2013 9:49	Archivo VOL	174 KB
	2011032803400300KDP.vol	12/07/2013 9:49	Archivo VOL	92 KB
	2011032803400300RhoHV.vol	12/07/2013 9:49	Archivo VOL	277 KB
	2011032803400300ZDR.vol	12/07/2013 9:49	Archivo VOL	179 KB
	2011032803500300dBZ.vol	12/07/2013 9:49	Archivo VOL	173 KB

Fig. 3. Ejemplos de volúmenes del radar de la EEA Anguil.

El formato de los volúmenes fue definido por el fabricante del radar y cuenta con una sección XML (Extensible Markup Language) que almacena datos del contexto de la toma de datos y una sección de datos en formato binario con compresión que contiene el dato plano y también presenta una estructura al estilo XML [12] [13]. Esta situación impone una restricción al uso de herramientas estándares para manipular XML [13], por lo que el acceso a estos datos del radar se dificulta, teniendo que utilizar el software nativo provisto por el fabricante el cual es propietario y de un alto costo.

Ante esta situación se desarrollaron un conjunto de programas que permiten descargar los archivos, transformarlos a dos formatos estándar (ASCII y GeoTIFF) y procesarlos (Fig. 4). Para desarrollar este software se utilizó Python 2.7 y las librerías externas PyQt4, Numpy, lxml, Gdal y Pyodbc. Como IDE de desarrollo de software se utilizó Geany y la visualización de las imágenes se realizó con Quantum GIS. El software se puede ejecutar en plataforma Linux o Windows.

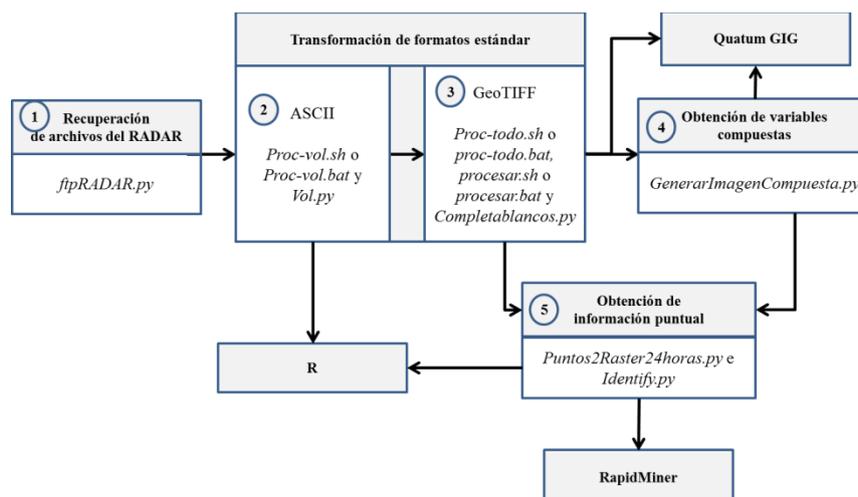


Fig. 4. Esquema de programas para el procesamiento de los datos del radar.

2.2. Preparar los datos.

El primer paso fue obtener los archivos del radar, para esto se usó el script de Python *ftpRADAR.py* indicando el radar a utilizar (Anguil), la serie de fechas (15 y 16 de Enero y 4,10,24 y 25 de Diciembre de 2012) y el rango en km (240). El script se conecta con el servidor del radar especificado y descarga los archivos de todas las variables registradas, que corresponden a la fechas del listado.

El segundo paso fue convertir estos archivos al formato estándar ASCII: con el proceso por lotes *Proc-vol* se recorre un directorio con archivos de datos crudos del radar y se pasan como parámetro al proceso *Vol.py*. Este script descomprime los volúmenes, transforma las coordenadas polares a geográficas, realiza el cálculo del dato “crudo” de las variables dBZ, PhiDP, KDP, ZDR y RhoHV; finalmente genera un archivo de texto por cada elevación con el formato: latitud, longitud y valor de la variable (Fig. 5). Si la toma de datos no tuvo inconvenientes, en un día completo (00:00 hs a 23:50 hs), se generan 8.640 archivos solo para el rango de 240 kilómetros.

```

1 lon lat dbz
2 -63.990067 -36.539683 49.500000
3 -63.992167 -36.543851 10.000000
4 -63.994267 -36.548019 17.500000
5 -63.996367 -36.552188 1.500000
6 -63.998467 -36.556356 0.000000
7 -64.000568 -36.560524 6.500000
8 -64.002668 -36.564692 12.500000
9 -64.004769 -36.568860 12.500000
10 -64.006871 -36.573029 9.500000
11 -64.008972 -36.577197 5.500000
12 -64.011074 -36.581365 8.000000
13 -64.013176 -36.585533 9.000000
14 -64.015278 -36.589702 7.500000
15 -64.017380 -36.593870 8.000000

```

Fig.5. Ejemplo del contenido de un archivo ASCII generado desde un volumen de la variable Z.

Estos archivos ASCII se leyeron y analizaron con el software R para verificar que los valores de las variables registradas por el radar estén dentro de los rangos esperados. Para el análisis se revisó la primera elevación en las 24 horas ya que al ser la más cercana a la superficie es la que mejor representa lo que puede precipitar a nivel del suelo [15] [16] [17]. Los datos corresponden a las 24 horas de todas las fechas seleccionadas para el análisis. Las variables presentan datos dentro de los rangos válidos con excepción de KDP (valores de -20,00 a -19,84 en todas las fechas cuando los valores válidos van de [-18; +36] [12]). En el caso de la variable PhiDP, los valores se encuentran dentro del rango permitido, pero el comportamiento de los mismos no es el esperado ya que los valores oscilan entre 0° a 1,39° cuando los valores normales durante una tormenta oscilan desde 0° a 360° [18]. Se encontró que el sensor del radar recibió la señal de forma incorrecta y en consecuencia tradujo erróneamente los valores en algunos de los archivos originales de ambas variables.

El tercer paso, es convertir los archivos de texto al estándar raster GeoTIFF, el cual permite trabajar los datos en Sistemas de Información Geográfica (SIG) y contar con una representación similar a las imágenes generadas por el software nativo de los radares. El formato raster representa adecuadamente los datos del radar con imágenes

de 1 km² de resolución, en coordenadas geográficas (latitud y longitud) y Datum WGS84. Este formato facilita la extracción de información de diferentes áreas del radar. Para generar las imágenes se utiliza *Proc-todo* que recorre el directorio de los archivos ASCII generados en el paso anterior y los pasa como parámetro al proceso por lotes *Procesar*. Este algoritmo recibe cada archivo de texto y genera una imagen de 487x505 píxeles. Como parte del proceso se ejecuta el script *Completablancos.py* el cual rellena todos los píxeles del área del radar interpolando el dato máximo correspondiente (Fig. 6).

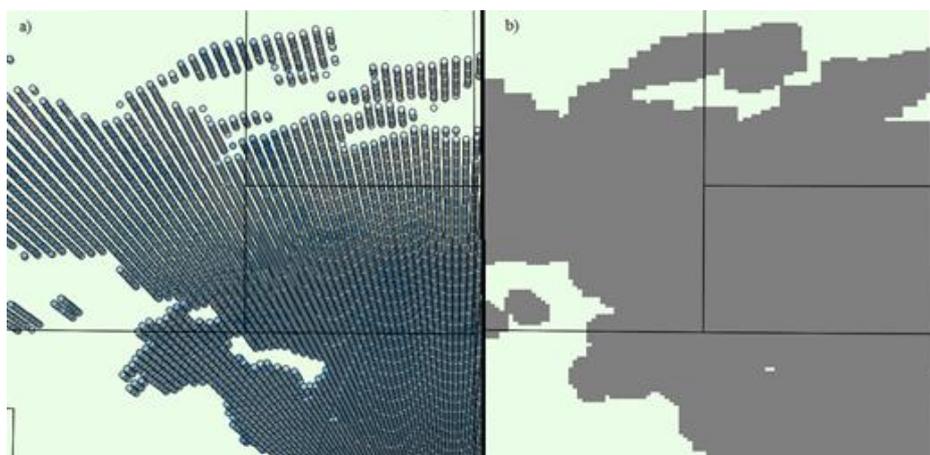


Fig. 6. a) Datos del radar en formato punto como están contenido en el archivo ASCII. b) Los mismos datos convertidos a formato raster donde cada dato ocupa un píxel de 1km²

El cuarto paso implica generar resúmenes con diferentes frecuencias temporales (cada 10 minutos y diarios) de las variables con diferentes cálculos (máximos y mínimos) y utilizando todas las elevaciones (perfil vertical de la tormenta) o solo algunas. El script *GenerarImagenCompuesta.py* calcula imágenes compuestas a partir de los argumentos indicados por el usuario (tabla 1). Para este trabajo se generaron imágenes de 24 horas para cada una de las fechas, con los valores máximos y mínimos de las variables Z, ZDR y RhoHV, para la primera elevación (Fig.7).

Tabla 1. Argumentos script *GenerarImagenCompuesta.py*

Argumentos posicionales:	
Path_img	Ubicación de los archivos raster a procesar para generar la imagen compuesta.
Fecha	Fecha a procesar, formato: aaaammdd
Extensión	Extensión de los archivos a procesar, el valor por defecto es "tif"
Variable a procesar	Posibles valores: dBZ, ZDR, RhoHV, KDP, PhiDP. Valor por defecto: dBZ
Argumentos opcionales:	
-h, --help	Muestra la ayuda
-mto	Calcula imágenes compuestas por cada paso

Argumentos posicionales:	
	de toma de datos (default: 10 minutos). Para todas las elevaciones.
-ele	Número de elevación a procesar. Posibles valores: 1 a 12. Si no se indica se procesan todas.
-d	Indica que hace falta generar las imágenes horarias para la imagen compuesta de 24 horas y todas las elevaciones.
-maxi	Genera la imagen compuesta con el valor máximo.
-mini	Genera la imagen compuesta con el valor mínimo

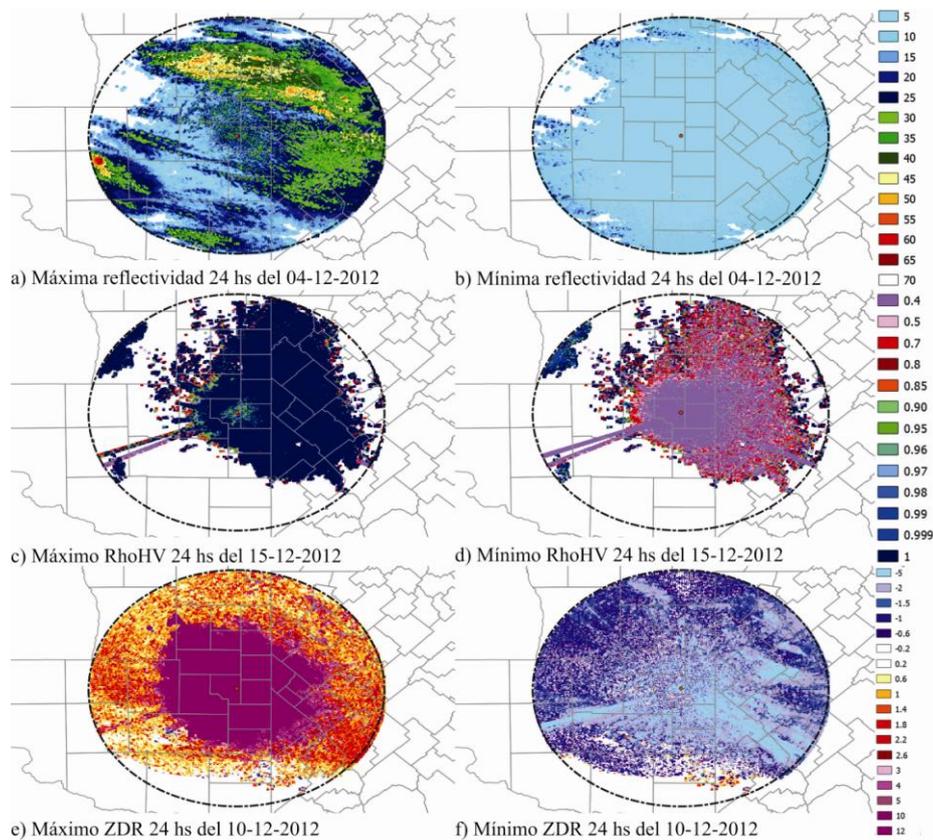


Fig. 7. Ejemplos de las imágenes compuestas de la primera elevación (0.5°) y 24 horas generadas con el script `GenerarImagenCompuesta.py` para tres fechas del 2012 con tormentas.

El quinto paso consiste en obtener los valores de las tres variables procesadas para una serie de puntos de verdad de campo de cada tormenta donde se indica si ocurrió granizo o no en esa localización. Para esta tarea se generó el script `Puntos2Raster24horas.py` que obtienen los valores del pixel correspondiente a cada par de coordenadas geográficas pasadas como parámetros (las cuales se pueden leer desde

una base de datos). La información recuperada se puede grabar en una base de datos o en un archivo ASCII. Utiliza el script *Identify.py* el cual convierte las coordenadas geográficas a coordenadas en pixel de la imagen raster pasada como parámetro y devuelve el valor correspondiente al pixel determinado. Para este trabajo se cuenta con **1279** casos (**494** son positivos y **785** son negativos) que se obtuvieron por medio de compañías de seguros agrícolas (SanCor, La Segunda y La Dulce), revisión de medios periodísticos nacionales y regionales, recorridas de campo y redes de Información (Red de Información Agropecuaria Nacional, Policía de La Pampa, Servicio Meteorológico Nacional, Agencias de Extensión de INTA).

El paso final de esta fase es el armado del dataset (tabla 2) para la generación de los modelos.

Tabla 2. Variables del DataSet preparado para la fase de modelar

Nombre	Tipo	Descripción
MxDBz1	Float	Contiene el máximo valor de reflectividad ocurrido durante las 24 horas en la primera elevación (Fig.7).
MxZDR1	Float	Contiene el máximo valor de ZDR ocurrido durante las 24 horas en la primera elevación. (Fig.7)
MxRho1	Float	Contiene el máximo valor de RhoHV ocurrido durante las 24 horas en la primera elevación. (Fig.7)
MnDbz1	Float	Contiene el mínimo valor de reflectividad ocurrido durante las 24 horas en la primera elevación. (Fig.7)
MnZDR1	Float	Contiene el mínimo valor de ZDR ocurrido durante las 24 horas en la primera elevación. (Fig.7)
MnRho1	Float	Contiene el mínimo valor de RhoHV ocurrido durante las 24 horas en la primera elevación. (Fig.7)
Latitud	Float	Ubicación espacial del dato. No se utiliza como variable de entrada.
Longitud	Float	Ubicación espacial del dato. No se utiliza como variable de entrada.
IdEvento	Int	Evento al que pertenece el dato. No se utiliza como variable de entrada.
Granizo	Bit	Target binario:0 no ocurrió granizo, 1 si ocurrió granizo

2.3. Modelar y Evaluar

El objetivo del modelo a construir es poder determinar a partir de las variables de doble polarización registradas por el radar si una tormenta presentó ocurrencia de granizo en superficie. Este es un problema de clasificación cuya la variable objetivo toma la forma $y \in \{0,1\}$, donde el cero corresponde a la clase negativa y el uno a la clase positiva. Para modelarlos se utiliza la regresión logística.

En el modelo de regresión logística se trata de calcular la probabilidad en la que una de las opciones de la variable y ocurra a partir de los valores que tomen una serie de variables independientes x . La hipótesis se representa por (1)

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n \quad (1) \quad [19] \quad [20]$$

Para que la función cumpla con la restricción de $0 \leq h_{\theta}(x) \leq 1$ se utiliza la función sigmoïdal, por lo que la función de hipótesis queda como (2).

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n}} \quad (2) \quad [19] \quad [20]$$

Para obtener los valores de los coeficientes (θ) de los predictores que mejor los relacionan a la variable objetivo, la regresión logística utiliza la estimación de la máxima verosimilitud. La función de costo a minimizar queda dada por (3).

$$J_{\theta} = -\frac{1}{n} \left[\sum_{i=1}^n y_i \log h_{\theta}(x_i) + (1 - y_i) \log(1 - h_{\theta}(x_i)) \right] \quad (3) \quad [19] \quad [20]$$

Una vez obtenidos los valores de θ , el resultado que arroja la función $h_{\theta}(x)$ determina la probabilidad que tiene el caso estudiado de pertenecer a la clase positiva, en general se predice $y=1$ si $h_{\theta}(x) \geq 0.5$ e $y=0$ si $h_{\theta}(x) < 0.5$.

Para generar el modelo se utilizó el software R. Como primer paso se realizó la separación del dataset en dos, una para entrenamiento (70% de los casos) y el otro para testeo (30%) de los casos.

Código R para preparar los dataset para la fase de modelado, generación del modelo, impresión de resultados y cálculo del error.

```
%Leo los datos del portapapeles y determine los valores
nulos

dsR <- read.table("clipboard", header=TRUE, sep="",
na.strings="9999", dec=".", strip.white=TRUE)

dsR[dat==99] <- NA
dsR[dat=='NULL'] <- NA

%Transforma la variable target en un factor
fix(dsR)

dsR$Granizo <- factor(dsR$Granizo, labels=c('No','Si'))

%Genero los dataset de entrenamiento y testeo
estratos <- strata( dsR, stratanames = c("Granizo"),
size = c(345, 549), method = "srswor" )

dsPRTr <- getdata(dsR,estratos)
```

```

library(sqldf)

dsPRTt = sqldf("select * from dsR where IdLocXEvento
not in (select IdLocXevento from dsPRTr)")

modelo <- glm(Granizo ~ MxDbz1 + MxRho1 + MxZDR1 +
MnDBZ1 + MnZDR1 + MnRho1, data = dsPRTr, family =
"binomial")

summary(modelo)

proba <- predict(modelo, newdata=dsPRTt,
type="response")

pred1 <- ifelse(proba <0.5, 0, 1)

mc <- table(dsPRTt$Granizo, pred1)

print(mc)

error <- 1.0 - (mc[1,1]+mc[2,2])/sum(mc)

print(error)

```

Posteriormente se ejecuta el commando glm indicando la familia binomial para determinar que es una regresión logística. Los resultados obtenidos presenta como variables significativas para el modelo MxDbz1, MxZDR1, MxRho1 y MnZDR1. El -2LL es igual a 627.42 y el AIC: 641.42. La matriz de confusión sobre testing arroja un error del 22,4%.

Se vuelve a realizar una corrida del modelo con las cuatro variables que resultaron significativas y el -2LL resulta en 624.47, el AIC en 634.47 y la matriz de confusión presenta un error de 23,6%. La variable MxRho1 aparece como no significativa.

El último modelo generado se realiza con las variables MxDbz1, MzZDR1 y MnZDR1. Todas las variables resultan significativas y -2LL es igual a 630.33, AIC: 638.33 y el error 23,6%.

No se ven diferencias importantes en el rendimiento de los tres modelos obtenidos, teniendo en cuenta el principio de parsimonia se selecciona el último que presenta la menor cantidad de variables.

La ecuación de la regresión del modelo final se presenta en (4)

$$P(Y=1) = \frac{1}{1 + \exp(-5.92548 + 0.14231xMxDbz1 - 0.17243xMxZDR1 - 0.37529xMnZDR1)} \quad (4)$$

Los signos de los coeficientes son coherentes con la bibliografía: a mayor reflectividad (MxDbz1) y menores valores de ZDR (MxZDR1 y MnZDR1), aumenta la probabilidad de la ocurrencia de granizo.

2.4. Implementación

Con la ecuación de la función logística, la misma se implementó en un script de Python *ClasificaciónRL.py*. Este script recibe el directorio donde están las imágenes de la tormenta y la fecha a clasificar. Recorre todos los píxeles de la imagen obteniendo los valores de las variables y aplicándoles la ecuación (4). Luego almacena dos imágenes, una con una clasificación con el corte en 0.5 y otra con los resultados de las probabilidades calculadas, con el objetivo que el usuario pueda seleccionar su propio valor de corte, para clasificar un caso como positivo o negativo.

Para probar el modelo se clasificó la tormenta del 15 de Enero del 2011, estos datos no fueron utilizados ni en el entrenamiento ni en el testeo del modelo. La figura 8 muestra la imagen clasificada con un valor de corte de 97%, en color rojo se ven los píxeles clasificados como positivos y en verde los negativos. Los puntos de colores rojos corresponden a datos de verdad de campo de ocurrencia de granizo y los verdes de no ocurrencia.

En la imagen se puede apreciar que la clasificación del modelo es bastante adecuada y se corresponde con los datos de campo, respetando el error de un 20% aproximadamente.

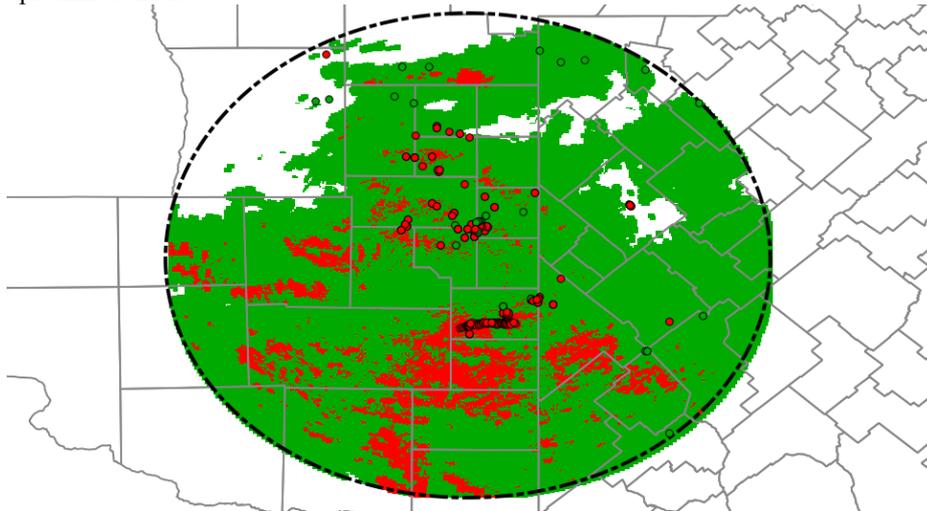


Fig. 8. Clasificación de la tormenta del 15 de Enero del 2011 utilizando el modelo generado en R con regresión logística.

Resultados

Se pudo aplicar con éxito el proceso CRISP-DM utilizando herramientas libres exclusivamente. Las herramientas desarrolladas con Python permiten llevar adelante las fases de preparación de datos al contemplar la descarga de información de los radares, la transformación de cinco variables (Φ DP, KDP, ρ HV, ZDR y Z), en el rango de 240 km, del formato propietario a dos formatos estándar: ASCII y GeoTIFF, la generación de imágenes de resumen y la obtención de datos puntuales de dichas

imágenes. En esta fase también se utilizaron herramientas libres de SIG como Quantum GIS para el tratamiento de las imágenes y R para el análisis de los datos.

Para las etapas de modelar y evaluar se utilizó el software estadístico R, obteniendo un modelo muy sencillo (solo tres variables) y con un muy buen rendimiento (24% de error). Los comandos utilizados en R, se pueden reutilizar para volver a entrenar el modelo o generar otros modelos con más casos y/o más variables, realizando cambios mínimos.

La fase de implementación se programó utilizando nuevamente Python, generando una manera sencilla de aplicar la regresión logística del modelo obtenido a nuevos datos, no utilizados para el entrenamiento y testeo del modelo.

Todas las herramientas desarrolladas y utilizadas se pueden ejecutar en plataforma Linux y Windows.

Estos resultados contribuyen a los objetivos de la red de radares de minimizar el uso de software propietario y brindar herramientas que permitan un mayor aprovechamiento de los datos de los radares.

Generan un marco de trabajo para la explotación de datos de los radares utilizando formatos estándares y abiertos que facilitan la interacción tanto entre las herramientas de software elegidas como entre los usuarios que las utilizan y garantizan la transparencia en los métodos utilizados (procesamientos y modelos) y la posibilidad de extender y mejorar las herramientas generadas.

Los trabajos futuros incluyen:

- Sobre las herramientas de procesamiento de la información de los radares: descarga de datos crudos en los rangos 120 y 480, transformación de más variables y más rangos, agregar más cálculos con las imágenes compuestas (ej: promedio).
- Sobre el análisis de granizo: la generación de modelos para la clasificación de granizo utilizando más variables de doble polarización, que incluyan el perfil vertical de la tormenta, predecir el daño en el cultivo con una variable objetivo categórica (Leve, Moderado, Severo) y otra continua (porcentaje de daño).
- Para ambos: optimizar la velocidad de procesamiento vectorizando aquellos cálculos que se puedan, pruebas de otras herramientas libres para modelar, evaluar e implementar (ej: RapidMiner, Octave).

Acknowledgments. agradecemos a la EEA Anguil por la inclusión en el programa de posgrados de INTA para la maestría en Explotación de Datos y Descubrimiento del Conocimiento de la Universidad Austral, en el marco de la cual se realizó este trabajo. También a la Coordinación de Información Agropecuaria Nacional, al Instituto de Clima y Agua, al proyecto territorial “PAMSL-1282101- Contribución al desarrollo territorial del este de La Pampa” del INTA y a las aseguradoras SanCor Seguros, La Segunda y La Dulce por el apoyo e información brindados para realizar este trabajo.

Referencias

1. Prensa Subsecretaria de Recursos Hídricos de la Nación. Argentina avanza hacia un plan de radarización meteorológica. (2012). http://www.hidricosargentina.gov.ar/novedad_interna.php?novedad=2414
2. A. Rodríguez, H. Ciappesoni. Sistema Nacional de Radares Meteorológicos. SINARAME. Simposio de Radarización y Sistemas de Alerta Hidrometeorológicas del Cono Sur. Buenos Aires, Argentina, (2012). http://pampero.cima.fcen.uba.ar/SimposioAlertas/docs/1_SINARAME.pdf
3. P. A. Mercuri, «Radares en red. seguimiento online de precipitaciones.», INTA Informa, n.o 576, mar. 2010.
4. M. Ceperuelo Mallafré, «Identificación y caracterización del granizo mediante el radar meteorológico. Modelos de predicción del ciclo de vida de las células convectivas.», Tesis Doctoral., 2008.
5. R. E. Rinehart, *Radar for meteorologists*. Dept. of Atmospheric Sciences, Center for Aerospace Sciences, University of North Dakota, 1997.
6. «Instruction Manual. Rainbow 5. Products and Algorithms.», Gematronik, 2007.
7. V. N. Bringi y V. Chandrasekar, *Polarimetric Doppler weather radar: Principles and applications*. Cambridge Univ Pr, 2001.
8. R. N. Mezher y P. A. Mercuri, «Uso de la red de radares de INTA para la detección de granizo», *XIII Reunión Argentina y VI Latinoamericana de Agrometeorología.*, oct. 2010.
9. D. L. Olson y D. Delen, *Advanced data mining techniques*. Springer Verlag, 2008.
10. R. Nisbet, J. Elder, J. F. Elder, y G. Miner, *Handbook of statistical analysis and data mining applications*. Academic Press, 2009.
11. «Gematronik Weather Radar Systems.» Selex Systems Integration GmbH, 2010
12. «Instruction Manual. Rainbow 5. File Format.», en *Instruction Manual. Rainbow 5.*, Gematronik, 2007.
13. T. Hartmann, M. S. Amburrino, y F. Bareilles, «Análisis preliminar de datos obtenidos por la red de radares del INTA para el estudio de precipitaciones en la Región Pampeana», 39 Congreso Argentino de Agroinformática. 2. Jornadas Argentinas de Informática. JAIIO, sep. 2010.
- 14.
15. K. Aydin, T. A. Seliga, y V. Balaji, «Remote sensing of hail with a dual linear polarization radar», *Journal of Climate and Applied Meteorology*, vol. 25, n.o 10, pp.
16. R. N. Mezher, S. Bancho, y Y. N. Bellini Saibene, «Identificación de granizo con la utilización de variables polarimétricas de los radares de Paraná y Anguil, el radar de Pergamino y daño en cultivos.», en Congreso Argentino de Meteorología. 11. 2012 05-06 28-01, 28 de mayo al 1 de junio de 2012. Mendoza. AR., 2012.
17. R. N. Mezher, L. Vidal, y P. Salio, «Hailstorms Analysis using Polarimetric Weather Radars and Microwave Sensors in Argentina», *6th European Conference on Severe Storms (ECSS 2011)*, 26082011.
18. R. J. Doviak y D. S. Znić, *Doppler radar and weather observations*. Courier Dover Publications, 1993.
19. D. W. Hosmer y S. Lemeshow, *Applied logistic regression*, Second., vol. 354. Wiley-Interscience, 2000. pp 15,31-32
20. A. Ng. Logistic Regression. Clasification. Lecture 6. Machine Learning Course. Stanford University. 2014. <https://d396qusza40orc.cloudfront.net/ml/docs/slides/Lecture6.pdf>