Universidad de la República

Facultad de Ingeniería

Reconocimiento de Patrones

INFORME DEL PROYECTO

Gastón García 4.595.401-2
g.garciagonzalez50@gmail.com
Henry Marichal 4.862.364-6
hmarichal@fing.edu.uy

December 6, 2016

Contents

1 Resumen

En el siguiente informe se trabaja con la detección de consumos eléctricos anómalos. Este problema se trata como un sistema de dos clases desbalanceadas. Para combatir el desbalance, se plantean distintas estrategias como: sobremuestreo de la clase minoritaria, submuestreo de la mayoritaria y pesos por clase. Se alcanza el mismo resultado en todas ellas.
Se evalúa el desempeño de los clasficadores: *Random Forest, SVM, K-Neighbours y Logistic Regression* individualmente como en conjunto, aplicando una estrategia de fusión. Se llega a que estos clasificadores no son complementarios ya que se comprueba que clasifican erróneamente el mismo conjunto de muestras.
Se plantea encontrar descriptores relacionados con la ubicación geográfica del cliente, dividiendo el espacio geográfico en grillas de distinto tamaño. Se comprueba que la transformación de características de ubicación geográfica no mejora significativamente la performance del clasificador *Random Forest*.
Para evitar el problema de 'la maldición de la dimencionalidad', se utilizan los métodos de selección de características: Selección de características de sub grupos basados en correlación, PCA y ponderación de características de *Random Forest*.

2 Objetivos

El objetivo principal de este proyecto es poder extraer características relacionadas con la ubicación geográfica tratando de mejorar los resultados de la detección de anomalías utilizando los consumos y las características nominales.

3 Descripción del problema

3.1 Introducción

El uso irregular o fraudulento de la energía eléctrica representa un problema que provoca cuantiosas perdidas económicas a las empresas distribuidoras de muchos países. En la red eléctrica existen los siguientes tipos de perdidas: técnicas (debidas a la disipación térmica-efecto Joule-), no técnicas (defectos en la medida de consumo, errores en al facturación, manipulación fraudulenta de equipos) y perdidas asociadas a las zonas carenciadas. Actualmente en Uruguay el total de las perdidas ronda alrededor del 20% de la energía generada.

Hace años se viene trabajando en la disminución de las perdidas no técnicas debido a la manipulación fraudulenta de equipos de medidas, las cuales rondan en un entorno del 4% del total de la energía generada.

La empresa UTE realiza la detección de consumos anómalos (no técnicos) en base a inspecciones al cliente (visitas). La fuente de generación de inspecciones a clientes sospechosos de fraude son : denuncias de terceros, personas ajenas a UTE que realizan la denuncia de situaciones irregulares; denuncias realizadas por personal de UTE; y por ultimo son las inspecciones llevadas a cabo en base a sospechas en el consumo eléctrico. Estas últimas son generadas por los ingenieros del área. Ademas son la mayor fuente de generación de Inspecciones, rondando entorno al 70% del total. Dentro de los criterios utilizados para definir clientes como sospechosos se destacan los siguientes: caída del consumo mayor al 40% de la media móvil semestral, sin inspecciones en 10 años, consumo nulo o varianza pequeña y reincidentes.

Este problema esta dentro de un mas general, detección de anomalías. Este consiste en detectar muestras que se apartan del comportamiento general. Para poder atacar el problemas se trabajará con una base de datos etiquetados la cual se puede representar como un sistema supervisado de dos clases desbalanceadas, consumos anómalos y normales (los anómalos son los consumos irregulares).

3.2 Antecedentes

Este problema se vine trabajando desde hace unos años dentro del Instituto de Ingeniería Eléctrica. El primer antecedente es en 2008 donde el problema es introducido por Juan Pablo Kosut y Diego Alcetegaray como proyecto del curso Reconocimiento de Patrones [4]. En este trabajo, se contó con una base de datos con consumos pertenecientes a comercios como almacenes y autoservicios en un periodo de tres años, donde cada muestras se encontraba etiquetada como consumos sospechosos y no sospechosos. La forma en que se trabajó fue considerando el problema como de una sola clase donde los consumos etiquetados como sospechosos eran *outliers*.

Luego, en 2011, Federico Decia, Matias Di Martino y Juan Molinelli, retoman el problema tomándolo como proyecto de fin de carrera [5]. En este mismo se amplió la base de datos sobre la que se venia trabajando, y en diferencia con el proyecto anterior se trato el problema como dos clases desbalanceadas.

En ese mismo año, Diego Introini y Daniel Lena [6], dentro del curso de Reconocimiento de Patrones, le dan otra perspectiva al análisis buscando identificar *clusters* dentro de los consumos y tratar de de ver si a estos se les podía asociar el rubro de los consumidores.

En el 2012, Fernanda Rodriguez y Sebastián Castro [7], también dentro del curso Reconocimiento de Patrones, propusieron la inclusión de nuevas características no extraídas de los consumos para entrenar el clasificador. Algunas de estas características eran, potencia contratada, zona, antecedentes, entre varias.

En el año 2014, Alicia Fernandez, Federico Locumberry y Fernanda Rodriguez realizaron un estudio sobre este problema en el cual se evaluaron nuevas técnicas. [2]

En ese mismo año, Diego Acuña y Lucia Korenko [3], teniendo en cuenta los antecedentes que se tenía hasta el momento retoman el problema trabajando en base al clasificador RandomForest. Intentando mejorar su desempeño utilizaron selección y/o extracción de características y balanceo entre clases. Dado a que el desempeño de este era similar al de otros clasificadores utilizados en otras instancias, deciden hacer un cambio de enfoque aplicando técnicas de *clusterfing*, para resolver el problema de la variabilidad de las muestras. La realización y los resultados del proyecto de Acuña y Korenko, fueron tomados como puntos de referencia en la realización del proyecto que se describe en este informe.

Desde hace un tiempo y en la actualidad se viene trabajando en un proyecto en conjunto con la universidad y el sector productivo (UTE). En la actualidad el equipo dentro de este proyecto esta conformado por: Juan Pablo Kosut, Fernando Santomauro, Andrés, por parte de UTE y Alicia Fernández, Federico Lecumberry, Matías Di Martino, Pablo Massaferro, María Inés Fariello, Pablo Zinemanas, Juan José Tacón, Henry Marichal, Sergio Martinez, por parte del Instituto de Ingeniería Eléctrica.

4 Medidores de desempeño.

Para medir el desempeño de un clasificador existen distintas medidas: Taza de aciertos, Taza de error, Recall, Precisión y F-mesure entre otras.

Para explicar estas medidas se deben explicar como se manejaran en este proyecto los siguientes términos:

- TP: True Positive, muestras que fueron clasificadas como anómalos cuando efectivamente son anómalos.
- TN: True Negative, muestras que fueron clasificadas como normales cuando efectivamente son normales.
- FP: False Positive, muestras que fueron clasificadas como anómalos cuando en realidad son normales.
- FN: False Negative, muestras que fueron clasificadas como normales cuando en realidad son anómalos.

Medidas:

$$T.\ acierto = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

La taza de acierto es la cantidad de muestras bien clasificadas sobre la cantidad total de muestras.

$$T.\ error = \frac{FP + FN}{TP + TN + FP + FN} \tag{2}$$

La taza de error es la cantidad de muestras mal clasificadas sobre la cantidad total de muestras.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Precision es la porción dentro de las muestras clasificadas como anómalos que en verdad lo son.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Recall es la proporción dentro de las muestras que son anómalos que son bien clasificadas.

$$F - measure = \frac{(1 + \beta^2) * Recall * Precision}{\beta^2 * Recall + Precision} \tag{5}$$

F-measure, integra las dos últimas medidas, variando el parámetro β se modifica el peso que se le da a cada uno de los medidores. En este proyecto se utilizo el valor de $\beta = 1$, para darle el mismo peso a las dos medidas.

En este proyecto se le dio mayor importancia a las medidas de *Recall*, *Precision* y *F-measure* ya que en un problema de clases desbalanceadas las tazas no tiene mucho sentido.

5 Base

Se dispone una base de datos de 7768 clientes. De estos, 7124 están etiquetados como normales y 644 se encuentran etiquetados como anómalos. Se cuenta con 100 consumos mensuales para cada cliente, ademas de características propias del cliente (nominales):

- Cantidad de irregularidades
- Fecha de inicio de contrato vigente
- Fecha de la última inspección
- Fecha de renovación
- Porcentaje de lectura reales
- Potencia contratada
- Latitud de la ubicación del Predio
- Longitud de la ubicación del predio
- Estado del acuerdo de servicio
- Mora

5.1 Evaluación

Para evaluar los resultados se dividen los datos en dos conjuntos, uno de entrenamiento y otro de test. Esta división se realiza manteniendo la proporción entre las clases. Además el conjunto de test esta conformado por un 20% del conjunto original siendo de 1524 muestras. Por lo tanto el conjunto de entrenamiento esta conformado por 6214 muestras con 515 consumos anómalos.

Para evaluar los resultados del conjunto de entrenamiento se realizan 10 validaciones cruzadas en 10 folds. Una validación cruzada en 10 folds consiste en dividir el conjunto de datos en 10 subconjuntos, entrenar el clasificador con 9 subconjuntos y clasificar el décimo. Luego cambiar el subconjunto de test a clasificar por otro de los 9, y entrenar con los 9 restantes. Esto hasta clasificar los 10 subconjuntos. Finalmente se promedian los indicadores de desempeño obtenidos para cada clasificación de los subconjuntos. Diez validaciones cruzadas, es este procedimiento repetido 10 veces.

5.2 Período de consumo

Esta base esta conformada por clientes que no fueron inspeccionados desde 2005 hasta 2015. A priori parece razonable contemplar la posibilidad que el cliente halla podido cambiar de clase durante el periodo temporal del que se disponen datos. Para esto se propone comparar diferentes periodos de consumos, y quedarse con el que maximice la performance.

Por otro lado, hay que tener en cuenta, que las fechas de inspección en su mayoría se realizaron durante el 2015, en diferentes meses. Por lo tanto, las etiquetas son validas hasta esa fecha inclusive. Por lo que se propone tomar diferentes periodos temporales desde diciembre del 2014 hacia el pasado.

Tomando los consumos como características se comparan periodos de 12, 24, 36, 48 y 60 meses utilizando como clasificador *Random Forest* haciendo validación cruzada[1]. Los resultados se observan en la tabla 1

meses	12	24	36	48	60
F-m (%)	26.53± 1.95	27.36 ± 1.56	27.59±1.17	27.36±2.34	28.23±2.73

Table 1: Valor de F-measure para diferentes periodos de consumos

A partir de los resultados de la tabla 1 se optó por el conjunto de muestras que corresponde al periodo de 36 meses, ya que con este se obtiene uno de los valores mas altos de *F-measure* que se pudo conseguir, pero ademas es el que menos varianza tiene.

[1]Se explicara en la siguiente sección

Agregando a estos consumos las características nominales, se obtiene un conjunto de 46 características, el cual al haciendo nuevamente validación cruzada con *RandomForest* se pudo obtener un rango de valores de *F-measure* por encima del que se obtenía con solo los consumos. Como técnica de balanceo se utilizo pesos por clases.

F-measure (%)
32.26 ± 3.90

Table 2: Valor de *F-measure* a partir de los consumos junto con las características de los clientes

6 Preprocesamiento de datos.

6.1 Estandarización

Debido a que en el clasificador SVM se debe realizar productos internos al entrenar y clasificar, los cuales pueden dar números muy grandes o muy pequeños, es que se debe estandarizar los datos. Esto consiste en hacer una transformación lineal a cada característica, de forma que el valor máximo de esta alcanzado por una muestra sea 1 y el mínimo alcanzado por otra muestra sea 0.

6.2 Normalización

Esta es una transformación de los datos que consiste en: a cada característica de cada muestra se le resta la media de las correspondiente característica y se la divide por la varianza de la misma.

Esta transformación fue aplicada previamente al uso de PCA, dado que esta técnica de selección y extracción de características devuelve las direcciones de máxima varianza, no se quiere que la distancia influya en el resultado.

7 Clasificadores

En la siguiente sección se describen los distintos clasificadores que fueron probados para este proyecto:

7.1 Random Forest

Random Forest es un clasificador que combina distintos arboles de clasificación, entrenados con el mismo conjunto de entrenamiento, pero con un conjunto de características seleccionadas de manera aleatoria para cada árbol. La cantidad de características dentro de este conjunto (m), junto con la cantidad de arboles (n) son algunos de los parámetros del clasificador. Una vez construidos los árboles para la predicción de una nueva muestra, Random Forest hace lo siguiente: clasifica a esta con cada árbol y la etiqueta que tenga mas veces se repita sera la que se le asignará a la muestra.

Suponiendo que se tienen N muestras de entrenamiento con M características, se utiliza el siguiente algoritmo para construir cada árbol:

1. Se toman N muestras de manera aleatoria con remplazo del conjunto de datos original. Con las cuales se entrenará.
2. Se eligen $m << M$ características de manera aleatoria, las cuales serán utilizadas de a una para la división de cada nodo. Se selecciona aquella que maximice el decrecimiento en impureza.
3. Se crecen los arboles hasta la máxima extensión posible. No se realiza podado.

En [8] se muestra que la tasa de error depende de la correlación entre los arboles y de la fuerza[2] de cada uno de estos. La tasa de error mínima se obtiene cuando se encuentra un óptimo entre la mínima correlación entre los arboles y la máxima fuerza de cada uno de estos. Ademas dado que se puede utilizar tantos arboles como se quieran[3] sin riesgo de sobre-entrenamiento se tiene que el único parámetro para el cual Random Forest es sensible es **m**. Este valor puede estimarse utilizando la tasa de error **oob**.

[2] un árbol con baja tasa de error es un clasificador fuerte

[3] lo ideal es seleccionar la mínima cantidad de arboles que permiten operar en el punto optimo de funcionamiento

Esta tasa se calcula a partir de las muestras que no se utilizan para entrenar el árbol. No se utilizan todas las muestras ya que cada árbol se construye tomando muestras con remplazo del conjunto de entrenamiento original. Se estima que aproximadamente un tercio de las muestras totales no se utilizan en la construcción del árbol [8]. La tasa de error **oob** se obtiene al clasificar estas muestras. Además estas muestras se pueden utilizar para obtener una estimación *"online"* del error de clasificación al ir aumentando la cantidad de arboles utilizados. En la figura 1 se observan los resultados.

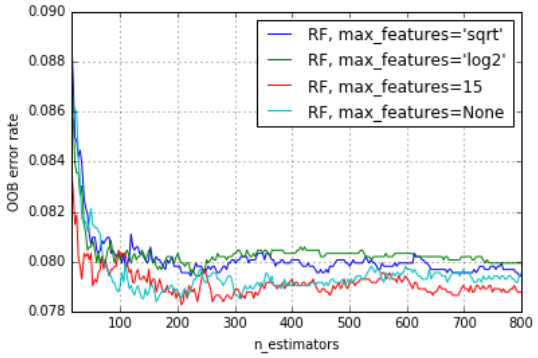

Figure 1: Tasa de error **oob** en función de la cantidad de arboles.

Lo ideal aquí es utilizar la mínima cantidad de características, que minimizan el error de clasificación y la minina cantidad de arboles a partir del cual el error de clasificación comienza a ser constante (menor costo computacional). En este caso a partir 700 arboles y 15 características[4]. **Oob** es una tasa de error de clasificación general por lo que para este problema en particular no es de interés minimizarlo. Además dado que la implementación utilizada no permite calcular la tasa de error de la clase minoritaria utilizando las muestras que no se utilizan en la construcción del árbol, para evitar el sobre-entrenamiento se utiliza validación cruzada.

Otro punto importante, es la posibilidad de utilizar este clasificador para seleccionar características. El mismo pondera las características según con que frecuencia son utilizadas para particionar los nodos, las de mayor frecuencia son las seleccionadas. Una estrategia para disminuir la cantidad de características sería, primero entrenar el árbol con todas las características y luego volver a entrenarlo utilizando solamente las características mas importantes.

Se utiliza como criterio de impureza la impureza de Gini.

Se esta trabajando, con un problema de dos clases desbalanceadas. Por lo que este clasificador tratará de minimizar la tasa de error general manteniendo baja la tasa de error de la clase mayoritaria mientras que la tasa de error de la clase minoritaria sera alta. Este desbalance puede ser contrarrestado utilizando diferentes pesos para las clases. Ademas se encontró que utilizando el criterio de parada de mínimos pesos por hoja se obtienen mejores resultados que sin utilizar criterio de parada.

7.2 SVM

Este clasificador, dadas las muestras de entrenamiento(x_i) con sus respectivas etiquetas(y_i), requiere la solución del siguiente problema de optimizacion:

$$\begin{array}{ll} \min\limits_{\text{w,b,}\zeta} & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\zeta_i \\ \text{s.a} & \begin{array}{l} y_i(w^T\phi(x_i)+b) \geq 1-\zeta_i \\ \zeta_i \geq 0 \end{array} \end{array} \tag{6}$$

Los vectores de entrenamiento son mapeados en un espacio de dimensión mayor por la función ϕ. Este clasificador busca un hiperplano de separación lineal con el margen[5] máximo en dicho espacio. Se define la función *kernel* como $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$. En este trabajo se utiliza RBF, $K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2}$.

El parámetro C, es común a todos los *kernel*. Este compensa las muestras mal clasificadas contra la simplicidad del hiperplano de decisión, es decir regular el tamaño del margen. Un bajo C, tiene como resultado

[4]para menor cantidad de arboles, si bien el error es mínimo hay mucha variabilidad(ruido)

[5]menor distancia entre la frontera de decisión y las muestras de entrenamiento

un mayor margen, mientras que un C alto tiene el objetivo de clasificar todas las muestras de entrenamiento correctamente (menor margen). El parámetro γ para el *kernel* RBF, define cuanta influencia tiene una muestra de entrenamiento (mayor sobre ajuste).

7.3 K-Neighbours

Este clasificador tiene un funcionamiento bastante sencillo, dado un conjunto de muestras de entrenamiento, cuando llega un muestra para ser clasificada, el clasificador, según una métrica preestablecida, detecta cuales son las K muestras de entrenamiento mas cercanas a la muestra de test y dentro de esas K muestras la etiqueta que se repita mas, sera la cual se le asignará a la muestra de test.

La herramienta que se utilizará para implementar este clasificador permite elegir distintas distancias variando el parámetro p de la distancia de *Minkowski*: $\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$. Esta herramienta también permite darle un mayor peso a los vecinos mas cercanas dentro de los K vecinos.

7.4 Regresión Logística

El modelo de regresión logística surge del deseo de modelar las posterior de cada clases, a través de funciones lineales en x, mientras que al mismo tiempo asegurarse de que estas sumen uno y permanezcan en el intervalo [0, 1]. El modelo es el siguiente:

$$\begin{cases} P(\omega_1|x) = \frac{e^{\omega^T \cdot x}}{1 - e^{\omega^T \cdot x}} \\ P(\omega_2|x) = \frac{1}{1 - e^{\omega^T \cdot x}} \end{cases} \tag{7}$$

Como se puede ver las posterioris suman 1 y ambas se encuentran en el intervalo [0,1]. Haciendo el cociente entre las posterioris y aplicándoles logaritmo se tiene:

$$\omega^T \cdot x = ln\left(\frac{P(\omega_1|x)}{1 - P(\omega_1|x)}\right) \qquad \text{(función logit)} \tag{8}$$

obteniéndose la función lineal en x que se pretendía, con el vector ω como incógnita. Con la base de entrenamiento se puede obtener este vector de forma iterativa, maximizando la verosimilitud:

$$p(t|\omega) = \prod_{n=1}^{N} P(\omega_1|x_n)^{t_n}(1 - P(\omega_1|x_n))^{1-t_n} \tag{9}$$

Donde t_n es la etiqueta de la muestra x_n.
Maximizar la ecuación 9 es análogo a minimizar la función de error $E(\omega)$:

$$E(\omega) = -ln(p(t|\omega)) \tag{10}$$

Buscando el ω que anule a ΔE, es que se llega a una ecuación implícita de ω, pudiéndose hallar a este de forma iterativa:

$$\omega(t+1) = \omega(t) + \eta\ (t_n - \omega(t)^T \cdot x_n)\ x_n \tag{11}$$

Una vez hallado el vector ω, se puede clasificar las muestras de test por la regla de decisión de Bayes (x pertenece a ω_1 si $P(\omega_1|x) > P(\omega_2|x)$ o de lo contrario pertenece a ω_2), lo cual por la ecuación 8 es lo mismo que hacer un producto escalar entre las muestras con ω y decidir en función del signo del resultado, entonces si el producto es positivo la muestra pertenece a la clase ω_1 y si es negativo pertenece a la clase ω_2, y el limite de decisión son los x tal que $\omega^T \cdot x = 0$ ($p(\omega_1|x) = p(\omega_2|x) = 0.5$).

8 Desbalance entre clases

Debido a que la cantidad de clientes normales es mucho mayor a la cantidad de clientes anómalos (aproximadamente una relación 11 a 1), se dice que hay un desbalance entre clases. Esto es un problema ya que esto puede producir un deterioro en la efectividad del clasificador que se vaya a usar, particularmente con las muestras pertenecientes a la clase minoritarias. El modelo va a estar sesgado a maximizar la cantidad de aciertos, esto implica clasificar todas las muestras como clientes normales, lo cual es un problema, porque todas las muestras de la clase minoritaria estarían siendo mal clasificadas.

Para tratar de mejorar esta situación es que se le aplicaron a la base a trabajar distintas técnicas de balance. El algoritmo utilizado para aplicar el desbanlanceo fue aplicar desbalanceo al conjunto de entrenamiento y clasificar el conjunto de test.

8.1 Submuestreo aleatorio de la clase mayoritaria.

Esta técnica consiste en eliminar muestras de la clase mayoritaria de manera aleatoria.

Relación	K-NeightBours	SVM	Random Forest	Logistic Regression
Sin Balanceo	20.94 ±5.32	7.49 ±6.50	19.03 ±5.82	8.36 ±6.43
10%	21.02 ±4.75	12.54 ±8.68	20.47 ±4.48	8.89 ±5.80
30%	27.10 ±4.76	27.72 ±4.89	30.87 ±3.18	20.59 ±3.70
50%	30.05 ±3.49	30.66 ±3.47	**32.00 ±4.39**	**29.82 ±2.55**
70%	**30.94 ±2.91**	**31.12 ±2.52**	29.36 ±2.93	29.29 ±3.57
90%	27.10 ±3.07	28.39 ±3.12	27.03 ±2.20	27.07 ±2.63

Table 3: Valor de F-measure de distintos clasificadores para distintas relaciones entre clases.

En la tabla 3 se muestran los valores de F-mesure de los distintos clasififcadores para distintas relaciones entre clases, donde 10% quiere decir que la relación es $\frac{anomalos}{normales} = 0.1$. Como se puede observar en esta tabla el desempeño mejora notoriamente para todos los clasificadores (en negrita se resaltan los mejores resultados) cuando se le aplica un balanceo a la base. Cabe observar también que las muestras minoritarias siempre se deben mantener por debajo de las muestras mayoritaria, en cuestión de cantidad, para obtener el mejor desempeño, ya que todos los clasificadores comparten que cuando la relación llega a 90% ($anomalos = 0.9 * normales$) el valor de F-mesure baja.

Para el resto de las técnicas de balanceo solo se representaran el mejor resultado para *Random Forest*

8.2 Sobre muestreo de la clase minoritaria

Se utilizan dos variantes de esta técnica. Una de ellas consiste en agregar copias aleatorias de las muestras de la clase minoritaria. Otra variante es generar muestras sintéticas. Para esta última se utiliza el algoritmo SMOTE (*Synthetic Minority Over -sampling Technique*), el cual para crear las muestras sintéticas, interpola entre dos o mas muestras similares (según cierta métrica).

A continuación se muestran los desempeño del clasificador *Random Forest* para ambas técnicas.

La mejor performance obtenida para la técnica de agregar copias, se consiguió para una relación de 0.85.

	Fvalue (%)
10 cv-10folds	31.98 ± 5.65
evaluación	29.6

Table 4: Valor de *F-mesure* del clasificador agregando copias.

Agregando muestras anómalas sintéticas, se llego a la mejor performance con una relación entre las muestras de la clase minoritaria y la mayoritaria de 0.99.

Como se puede apreciar en las tablas 4 y 5 para la copia de muestras se obtiene un mejor desempeño.

Una observación sobre este balanceo es que la optimización de la clasificación se da cuando las clases son prácticamente iguales en cantidad de muestras.

	Fvalue (%)
10 cv-10folds	29.1 ± 4.15
evaluación	28.9

Table 5: Valor de *F-measure* del clasificador con SMOTE

8.3 Pesos por clase

Las implementaciones de *Random Forest*, *SVM* y *Logisitic Regression* cuentan con la opción de asignar pesos por muestras y por clase. Utilizando esta última opción, el valor de F-mesure para *RandomForest* es:

	Fvalue (%)
10 cv-10folds	31.23 ±5.45
Test	30.3

Table 6: Valor de F-measure del clasificador

RandomForest, tiene la opción de asignar un costo de clasificación errónea para cada clase. La idea para balancear las clases es asignarle un mayor peso (costo) de clasificar de forma errónea las muestras minoritarias. A estos pesos el clasificador los usas en dos lugares, en los nodos de cada árbol para ponderar las divisiones de estos, incluyendo los nodos terminales u hojas para tomar la decisión final de cada árbol; y en el momento de hacer la votación entre todos los arboles para tomar la decisión final, el algoritmo multiplica la cantidad de votos de cada clase por su respectivo peso y la etiqueta de la clasificación corresponderá al mayor valor. En este caso se utilizo un peso de 7 para la clase minoritaria y de 1 para la mayoritaria. Además se utilizo un criterio de parada de mínimo peso por hoja de 0.02. Para obtener el peso por muestra, se divide el peso de la clase entre la cantidad de muestras de dicha clase.

El clasificador SVM, tiene la opción de variar para cada clase el valor con el que se debe multiplicar el parámetro C de SVM (explicado en la sección 7), pudiendo mover cada eje del umbral de forma independiente para cada clase. La forma de aplicar balanceo entre clases con SVM es, haciendo que el valor de multiplicación de C de la clase minoritaria sea mayor que el de la clase mayoritaria.

Al igual que RandomForest, LogisticRegression tiene la opción de asignar un costo de clasificación errónea para cada clase.

Una vez probadas las distintas técnicas de balanceo se puede ver que los todas las técnicas mejoran el desempeño de los clasificadores. Los máximos valores de *F-mesure* son similares para todas las técnicas, por lo cual se pueden usar cualquiera de ellas. de aquí en más se usara el submuestreo de la clase mayoritaria.

9 Voting

Este método combina una serie de clasificadores distintos para luego hacer un votación entre todas las decisiones, con la posibilidad de asignar distintos pesos a la decisión de estos. El procedimiento de clasificación una vez entrenados los clasificadores es: cada clasificador clasifica la muestra que llega, si todos los clasificadores tienen el mismo peso, la etiqueta asignada sera la que mas veces se repita entre todos los clasificadores; si los clasificadores tiene distintos pesos, se sumaran los pesos de una misma clase y el valor mayor obtenido sera el que tenga la posibilidad de etiquetar la muestra con su respectiva clase.

Los clasificadores que se combinaron en este proyecto fueron: RandomForest, K-Neighbors, SVM y Regresión logística. Como técnica de balanceo se utilizó una relación entre muestras de 0.5. La estrategia adoptada fue de Fusión, dado que los tres clasificadores trabajaron sobre el mismo espacio de características y que la elección de la etiqueta se dio por votación. El resultado obtenido fue el siguiente:

Dado a que no se notaron mejoras en el rendimiento, se observaron que muestras estaban siendo mal clasificadas, tanto los falsos positivos como los falsos negativos, y se noto que los cuatro clasificadores se equivocaban prácticamente en el mismo conjunto de muestras. Esto, junto con que se usó un numero par de clasificadores, puede explicar el menor rendimiento con respecto a cada clasificador por separado. También al compartir

	Fvalue (%)
10 cv-10folds	28.53 ±2.30
Test	28.02

Table 7: Valor de F-measure usando Voting.

prácticamente el mismo conjunto de muestras mal clasificadas, vimos que no tiene sentido aplicarle mas peso a un clasificador que a otro.

10 Extracción de características

En esta parte se trabajará con nuevas características y se comparará su desempeño con las anteriores.

10.1 PCA

En la búsqueda de mejorar el desempeño de los clasificadores que se vienen trabajando, se le aplico a la base de muestras un análisis de componente principales (PCA). Una de las interpretaciones de este algoritmo es proyectar los datos en las direcciones de mayor varianza. Este método puede utilizarse como un método de extracción de características y/o selección

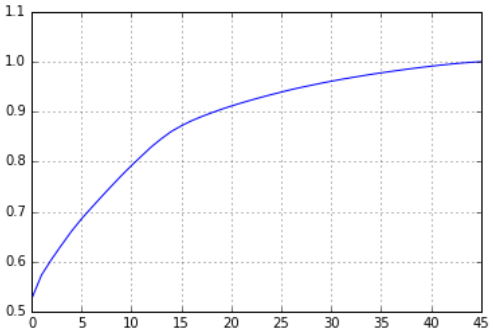

Figure 2: Varianza acumulada para PCA de los consumos junto con las características nominales

Observando la figura 2, se aprecia que para el conjunto de características conformado por los consumos y las características de los clientes, no hay componentes en las cuales se concentre en mayor medida la varianza. Por ejemplo, el 95% de la varianza se concentra en 30 características. Considerando que se tienen 46 características, es posible afirmar que no reduce de manera considerable la dimensionalidad.

El procedimiento utilizado para aplicar PCA correctamente consistió primero en normalizar[6] los datos de entrenamiento para luego aplicarles PCA. Al conjunto de test se le aplico las mismas transformaciones definidas por el conjunto de entrenamiento, es decir, para normalizar se resto la media y se divido por la varianza del conjunto de entrenamiento, y se aplico la transformación PCA definida por el mismo conjunto (vectores propios)

En la tabla 8 se observan los resultados obtenidos clasificando los datos variando la cantidad de componentes.

[6]media 0 y varianza 1

	F-measure (%)(cv)	F-measure (%)(evaluación)
10	30.36 ± 3.39	28.4
20	30.29 ± 3.03	27.8
30	29.43 ± 3.57	28.3
40	**29.71 ± 4.85**	**29**
46	28.59 ± 3.91	25.5

Table 8: Valor de F-measure para distintas cantidades de componentes principales

El mejor resultado se obtiene al utilizar 40 componentes principales. Sin embargo, este resultado es ligeramente menor al obtenido sin aplicar PCA (ver tabla 2). Este puede deberse a que los parámetros que se están utilizando para el clasificador son los determinados por el conjunto de entrenamiento sin transformar. Se hizo de esta manera, debido al costo computacional que conlleva optimizar los parámetros para cada conjunto. En la tabla 9 se observan los resultados al clasificar utilizando 40 componentes principales con los parámetros óptimos.

	Precisión(%)	Recall(%)	Fmeasure (%)
10-cv	23.51± 3.41	45.06 ± 4.56	30.81±3.69
evaluación	22.3	40.0	28.6

Table 9: Valores obtenido con los parámetros óptimos

Los resultados no mejoraron , aunque son muy similares a los obtenidos sin aplicar PCA(tabla 2).

10.2 Transformar características de ubicación geográfica.

En [1] se propone transformar las características de ubicación geográfica (latitud, longitud). La transformación propuesta es dividir el área en la que los clientes están localizados en grillas de distintos tamaños. Para cada celda de la grilla se calcula la proporción de clientes anómalos sobre la cantidad de clientes inspeccionados de dicha celda. En una primera instancia se utilizan grillas de los siguientes tamaños:,

	área por celda(km2)
5x5	27.8
8x8	10.8
10x10	6.94
15x15	3.09
30x30	0.770
50x50	0.278
100x100	0.0690
200x200	0.0174
300x300	0.00771

Table 10: El área total abarcada es de 694.323 km2

Ademas se propone explotar la información que brinda la ubicación geográfica mezclándolas con algunas características propuestas en trabajos anteriores[3]. Estas son:

1. Relación entre el consumo medio y el promedio de los consumos de los últimos 3 meses
2. Relación entre el consumo medio y el promedio de los consumos de los últimos 12 meses
3. Cociente entre la varianza del consumo del ultimo año respecto a la varianza promedio de todos los consumos
4. Pendiente de la recta que mejor se ajusta a la curva de consumos

5. Mora(días desde la ultima inspección)(característica nominal)

Las características que se proponen son:

6. Promedio de la característica 1 de todos los clientes de la celda
7. Promedio de la característica 1 de los clientes anómalos de la celda
8. Promedio de la característica 2 de todos los clientes de la celda
9. Promedio de la característica 2 de los clientes anómalos de la celda
10. Promedio de la características 3 entre todos los clientes de la celda
11. Promedio de la características 3 entre los clientes anómalos de la celda
12. Promedio de la características 4 entre todos los clientes de la celda
13. Promedio de la características 4 entre los clientes anómalos de la celda
14. Promedio de la características 5 entre todos los clientes de la celda
15. Promedio de la características 5 entre los clientes anómalos de la celda
16. Promedio de la media de los consumos de todos los clientes de la celda
17. Promedio de la media de los consumos de los clientes anómalos de la celda
18. Promedio de la varianza de los consumos de todos los clientes de la celda
19. Promedio de la varianza de los consumos de los clientes anómalos de la celda

Eliminando a los consumos como características y agregando estas nuevas se tiene el siguiente conjunto. Las características a partir de la 6(14 características) en adelante son características que dependen de la celda(14*9). Además se tienen 4 características extraídas de los consumos y las 10 características nominales. Por lo tanto la cantidad de características que se tienen son 4+14*9+10+9= 149.

Los resultados que se obtienen al clasificar las muestras de entrenamiento utilizando solamente las 9 características que indican la proporción de anómalos en la celdas, y como clasificador Random Forest son :

Precision(%)	Recall(%)	Fmeasure (%)
57.06± 5.20	97.73 ± 2.85	71.89±4.20

Table 11: Resultados al utilizar cv en 10 Folds con el conjunto de entrenamiento.

Estos parecen bastante prometedores. Sin embargo al observar los resultados sobre el conjunto de evaluación:

Precision(%)	Recall(%)	Fmeasure (%)
7.5	6.2	6.8

Table 12: Resultados de entrenar con el conjunto de entrenamiento y clasificar al conjunto de evaluación.

Al parecer se produjo sobre-entrenamiento. Una posibilidad es que debido a que el tamaño de algunas celdas es demasiado pequeño, por lo que alguna de las características que indica la proporción de anómalos sobre el total pueda estar causando el problema. Si las celdas son lo suficientemente pequeñas para que en una parte importante se ubique un único cliente, el valor de dicha celda, va a ser el valor de la etiqueta. Si a la celda, pertenece un cliente anómalo, dicha característica tomara el valor 1, mientras que si el cliente esta etiquetado como normal, el valor de la etiqueta sera 0.

En la figura 3 se observa el gráfico entre las características y la importancia que Random Forest les da. La característica 9 es la que se determina con la grilla de 300x300. Como se esperaba, el clasificador les da un grado de importancia en orden creciente.

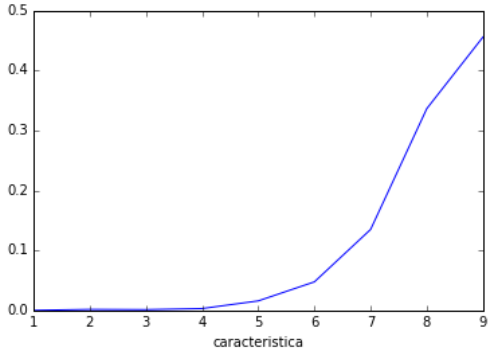

Figure 3: Ponderación de cada característica (RF)

Otra manera de verificar que este es el problema, es utilizando una unica grilla de 1000x1000. Los resultados obtenidos al evaluar con 10 cv al conjunto de entrenamiento de una unica caracteristica son:

Precision(%)	Recall(%)	Fmeasure (%)
62.09± 4.79	97.58 ± 2.16	75.74±3.71

Table 13: Resultados al utilizar cv en 10 Folds con el conjunto de entrenamiento de una característica

Como estrategia para solucionar el problema se plantea ir disminuyendo los tamaños de grilla, mientras se compara la performance obtenida entre cv-10Folds del conjunto de entrenamiento y la obtenida al clasificar el conjunto de test. Utilizando únicamente las características de proporción de clientes anómalos por celda para los tamaños de grilla 5x5,8x8,9x9 se obtuvo:

	Precisión(%)	Recall(%)	Fmeasure (%)
10-cv	28.41± 7.39	29.4 ± 6.13	28.71±6.32
evaluación	25.7	29.5	27.4

Table 14: Resultados de clasificar utilizando las tres características de proporción para las diferentes grillas.

Por lo que dado que el valor de F-measure obtenido para el conjunto de test se ubica en el entorno estimado por validación cruzada, se considera que se logro mitigar el sobre-entrenamiento.

Si consideramos a las características de proporción de clientes anómalos en la celda como las características transformadas a partir de la ubicación geográfica, puede ser pertinente compararlas. Para esto se observa la performance obtenida por el clasificador al utilizar como características solamente la latitud y longitud.

	Precisión(%)	Recall(%)	Fmeasure (%)
10-cv	21.89± 4.44	37.61 ± 4.17	27.59±4.48
evaluación	18.7	36.4	24.7

Table 15: Resultados clasificando solo usando Longitud y Latitud.

En la tabla 15 se pueden apreciar los resultados. Estos son ligeramente menores que los obtenidos en la tabla 14.

Resumiendo se tienen las 4 características extraídas de los consumos, 3 características de proporción de clientes anómalos por celda, 42 características derivadas de la ubicación geográfica y las características extraídas de los consumos, y con las características nominales de los clientes sin latitud y longitud. Esto hace un total de 57 características por lo que a merita utilizar algún método de selección para evitar el problema de la maldición de la dimensionalidad, ya que se tienen $\frac{515}{57} = 9$.

11 Selección de características

11.1 PCA

Como primer método de selección se utiliza nuevamente PCA. En la figura 4 se observan los resultados al aplicar dicho método. Utilizando mas de 30 componentes no se agrega varianza a los datos. Esto puede deberse a la correlación que hay entre las características.

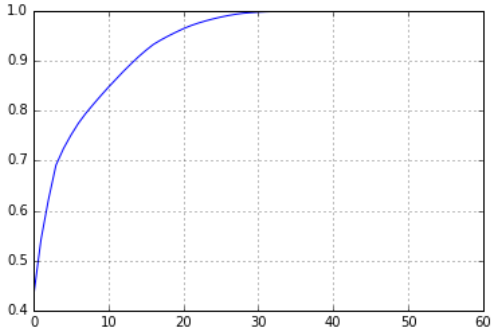

Figure 4: Varianza acumulada en función de las componentes principales, extraídas de las características de la ubicación geográfica, nominales y las obtenidas de los consumos

	Fmeasure (%)(cv)
10	32.46 ± 6.17
20	**32.48 ± 5.19**
30	31.96 ± 5.69
40	31.2 ± 5.98
50	29.7 ± 6.74
59	29.68 ± 5.54

Table 16: Valor de F-measure para distintas cantidades de componentes principales

En la tabla 16 se observan los resultados de clasificar los datos. Los resultados utilizando PCA son mejores que en el caso anterior(tabla 2) ya que utilizando menos características se llega a una performance ligeramente superior.

Precisión(%)	Recall(%)	Fmeasure (%)
29.41± 4.48	37.34 ± 3.55	32.80±3.79

Table 17: Valores obtenido con los parámetros óptimos

En la tabla 17 se observan los resultados utilizando parámetros óptimos del clasificador. Debido a estos resultados, utilizar PCA como un método de extracción/selección de características permitió buenos resultados ya que se paso de 46 características a 20($\frac{515}{20} = 25.75$).

11.2 Selección de subgrupos de características.

En esta subsección se utiliza el criterio de selección de características de subgrupos basado en correlación(*CfsSubsetEval*). Este criterio busca encontrar el subconjunto de características que presenta una mayor correlación entre las clases y que a la vez las muestras del subconjunto estén lo menos correlacionadas entre si. Para ello busca el subgrupo que maximiza la siguiente función de merito:

$$J_{cor}(S) = \frac{\sum_i U(x_j, C)}{\sqrt{\sum_i \sum_j U(x_j, x_i)}} \forall i, j \in S \quad con \quad U(X,Y) = \frac{2(H(X) - H(X|Y))}{H(X) + H(Y)} \tag{12}$$

El denominador representa la correlación media entre las características de S (sub conjunto de muestras) y las clases, mientras que el denominador es la intercorrelacion entre las características de S.

Para buscar el mejor subconjunto se utiliza la búsqueda *BestFirst*. La ventaja de este método sobre la búsqueda secuencial clásica es que no termina al dejar de mejorar el desempeño sino que mantiene registro de los mejores subconjuntos y regresa a considerar conjuntos previos. Como desventaja tiene que se incrementa el costo computacional entre 2 y 10 veces.

El subgrupo de características seleccionado es

$$CfsSubSetEval = [58, 55, 50, 48, 44, 31, 26, 21, 9, 6, 5, 2, 0]$$

Precisión(%)	Recall(%)	Fmeasure (%)
26.33± 3.03	46.84 ± 6.00	33.6±3.59

Table 18: Resultados utilizando las características seleccionadas

En la tabla 18 se tienen los resultados obtenidos. Estos son los mejores resultados, en términos de performance obtenidos hasta el momento.

11.3 Random Forest

Anteriormente se explico que random forest cuenta con un parámetro que permite saber cuales características utilizo con mayor frecuencia para realizar las divisiones de los conjuntos. En la figura 3 se observa la ponderación por características. Se observa que a las primeras 7 características les da bastante ponderación. Estas son las 4 extraídas de los consumos en conjunto con las 3 características de proporción de anómalos por celda. Luego les da bastante ponderación a las características nominales, entre ellas a la mora le da la mayor ponderación.

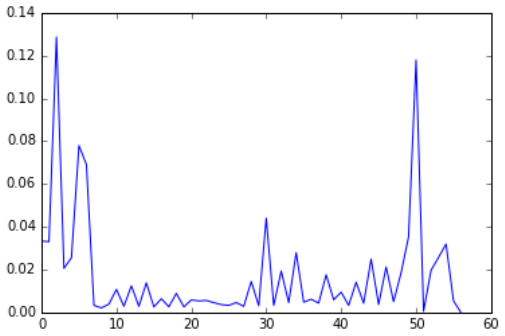

Figure 5: Ponderación de características Random Forest

ponderacion(RF)≥	Fmeasure (%)(cv)	evaluación (%)
0.01	32.61 ± 4.1	31,1
0.015	**33.69 ± 3.24**	**30.2**
0.02	33.6 ± 3.87	30.1
0.025	33.45 ± 3.82	29.2
0.03	32.88 ± 3.34	29.2
0.04	31.09 ± 3.83	30.0

Table 19: Resultados para los distintos umbrales

El mejor resultado se da cuando se utilizan las características que tienen un ponderación mayor a 0.015. Estas son un total de 19. Este resultado es el mejor de todos los obtenidos, aunque es muy similar al obtenido utilizando el subconjunto determinado por el criterio de correlación. Esto puede deberse a que el criterio que se esta utilizando esta directamente relacionado con el clasificador.

12 Conclusiones

Un de los grandes problemas a resolver en este proyecto fue el desbalance entre clases. Para solucionar este problema se utilizaron técnicas de balanceo como: Submuestreo, Sobremuestreo, y ademas se aplicaron pesos por clase en los clasificadores que lo permitían. Para todos los casos se alcanzaron resultados similares, por lo que se considera resuelto el problema del desbalanceo, ya que con técnicas diferentes se alcanza el mismo máximo en performance del clasificador.

Después de probar con distintos clasificadores como: *Random Forest*, *SVM, K-Neighbours* y *Logistic Regression*, utilizando los consumos y las características nominales, y al ver que todos tenían desempeños similares, se planteó la posibilidad de usar *Voting*. Para esto se miraron los conjuntos mal clasificados de cada clasificador para ver si tenia sentido que los clasificadores en conjunto levantaran la performance. Comprobándose que todos los clasificadores se equivocaba en un similar conjunto de muestras. Esto se vio reflejado cuando se aplico la herramienta y se obtuvo un resultado ligeramente inferior.

En cuanto a la extracción de nuevas características relacionadas con la ubicación, no se obtuvo una mejora considerable. Ademas utilizando que PCA se puede interpretar como un cambio de base, observando que en las 30 primeras componentes se acumulaba el 99.9% de la varianza, se observó que el total de las características propuestas tienen una gran correlación. Esto tiene sentido ya que todas estas contienen información de la ubicación.

También se quiere destacar, la versatilidad de la herramienta *Random Forest*, ya que esta cuenta con la posibilidad de, contemplar el desbalance entre clases, hacer selección de características, clasificar y evaluar (oob).

References

[1] Patrick Galuner,Jorge Meira, Lautaro Dolberg, Radu State, Frack Bettinger y Yver rangoni, *Neighborhood Features Help Detecting Electricity Theft in BigData Sets*

[2] Fernanda Rodríguez, Federico Lecumberry y Alicia Fernández, *Proyecto CSIC Sector Productivo Modalidad I UTE Detección de registros de consumo Anómalos*

[3] Diego Acuña y Lucia Korenko, *Proyecto Reconocimiento de Patrones 2014: Detección de consumos anómalos de energía eléctrica*

[4] Juan Pablo Kosut y Diego Alcegaray, *Proyecto Reconocimiento de Patrones 2008: One Class SVM para la detección de fraudes en el uso de energía eléctrica*

[5] Federico Decia, Matias Di Martino y Juan Molinelli, *Proyecto de grado 2011: Detección de consumos anómalos (De-Ca)*

[6] Diego Introini y Daniel Lena, *Proyecto Reconocimiento de Patrones 2011: Proyecton detección de clusters*

[7] Sebastián Castro y Fernanda Rodriguez, *Proyecto Reconocimiento de Patrones 2012: Detección de consumos anómalos energía eléctrica utilizando nuevas características*

[8] L.Breiman, *"Random Forests", Machine Learning, 45(1), 5-32, 2001*