Reconocimiento de Patrones

Edición 2017

Informe Proyecto Final

Gonzalo Marín[‡]

2 de diciembre de 2017

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería
UDELAR

[‡]ing.gonzalo.marin@gmail.com

Índice

1. Introducción

Este trabajo se enmarca en un proyecto de la gerencia técnica de Plan Ceibal sobre caracterización del tráfico cursado en la red. Parte de los objetivos de dicho proyecto son: comprender el comportamiento individual de los beneficiarios al hacer uso de la conectividad y encontrar patrones de comportamiento. Esto permitirá dar respuesta a preguntas tales como: ¿qué dispositivos tienen los usuarios más activos?, ¿cuánto tiempo usan la red y cómo se distribuye el uso a lo largo del día y la semana?, ¿están concentrados en cierto nivel escolar, socio-cultural, lugar geográfico?, ¿qué aplicaciones usan quienes más tráfico cursan?, entre otras.

La metodología de aprendizaje utilizada para la caracterización de usuarios está basada en aplicar algoritmos de clustering no supervisado a partir de un vector de características diseñado para dicho fin. Como método de validación, se propone analizar el poder de discriminación que las características seleccionadas tienen, resolviendo un problema de clasificación supervisada con los datos disponibles. Esto último es lo que motiva a este trabajo, y es de medular importancia para justificar que la segmentación que se realice a posteriori tiene significado respecto a la caracterización de usuarios.

Todo el procesamiento y análisis de los datos fue realizado mediante *scripts* en lenguaje `Python 3.6`. Se entregan adjuntos a este informe dos archivos: un `notebook jupyter` que contiene todo el desarrollo y código de implementación, y el conjunto de datos utilizado (ver Anexo A).

El presente informe se distribuye de la siguiente forma: en las Secciones 2 y 3 se presentan los objetivos y se describe el conjunto de datos a utilizar. En la Sección 4 se presenta una primera aproximación al problema de clasificación. En la Sección 5 se estudia un problema particular de desbalance de clases. Para finalizar, en la Sección 6 se realiza la comparación entre los resultados obtenidos mediante un análisis de incertidumbre y en las Secciones 7 y 8 se presentan las conclusiones y trabajo futuro, respectivamente.

2. Objetivos

El objetivo principal de este trabajo es estudiar si a partir de datos de tráfico, es posible identificar al tipo de dispositivo utilizado para navegar, según las siguientes categorías:

1. laptop entregada por Ceibal
2. tablet entregada por Ceibal
3. dispositivo personal (no entregado por Ceibal)

Para ello, se propone resolver un problema de aprendizaje supervisado, utilizando un vector de características que fue diseñado con el fin de realizar caracterización de usuarios. Si dichas características brindan buenos resultados respecto a la clasificación, esto será un buen indicio sobre su poder de discriminación. De esta forma, podrá ser utilizado (a futuro) para realizar la segmentación en grupos de usuarios mediante clustering no supervisado.

Si bien resolver el problema de clasificación es el objetivo central del presente trabajo, también se pretende responder a las siguientes preguntas:

1. ¿Es posible inferir a partir de los datos de tráfico si el dispositivo es de los que entrega Ceibal? Una respuesta afirmativa a este punto indicaría que el patrón de tráfico es distinto según si el dispositivo es entregado por Ceibal o si se trata de un dispositivo personal.

2. ¿Es posible identificar dentro de los dispositivos que entrega Ceibal si se trata de una tablet o una laptop? En particular, una respuesta afirmativa a este punto indicaría que el tráfico cursado por un usuario con una laptop es distinto al cursado con una tablet.

3. Construcción del conjunto de datos

3.1. Fuente de datos

El conjunto de datos utilizado para este trabajo fue construido a partir de datos de monitoreo del uso de la red Ceibal, recolectados con el software NTOP[1] en aproximadamente 100 centros educativos (que incluyen escuelas públicas, liceos públicos y UTUs). Cada registro del conjunto de datos corresponde al tráfico cursado (*downlink* y *uplink*) por un dispositivo conectado a la red Wi-Fi de Plan Ceibal, diferenciado por aplicación (YouTube, Facebook, Google, etc.). La base de datos cuenta con la siguiente información:

- `TIEMPO_INICIO`: timestamp que representa el inicio de la ventana de tiempo de estudio.
- `MAC`: dirección `MAC` del dispositivo. A partir de este elemento es que se distingue si se trata de un dispositivo entregado por Ceibal (laptop o tablet) o si se trata de un dispositivo personal.
- `APLICACIÓN`: la aplicación a la que corresponde esta entrada.
- `DOWNLINK`: tráfico de bajada en Bytes, medido en la ventana de tiempo indicada en el campo `WINDOW`.
- `UPLINK`: tráfico de subida en Bytes, medido en la ventana de tiempo indicada en el campo `WINDOW`.
- `LOCAL`: centro educativo al que corresponde esta entrada.
- `WINDOW`: tamaño de la ventana de análisis. Este valor está fijo en 300 segundos.

3.2. El vector de características (*features*)

El vector de características fue diseñado a partir del conjunto de datos disponible. Este se centra en dos aspectos principales del uso de la red:

1. volumen de tráfico cursado
2. tiempo de uso

Para ambos casos, se definieron histogramas que representan la distribución de los datos según el siguiente conjunto de filtros:

- Franja horaria: de 7:30 a 17:30 hs.

[1] http://www.ntop.org/

- Aplicaciones:
  - 15 no Ceibal y 5 Ceibal (20 en total, las de mayor uso), solo *downlink*.
  - 15 no Ceibal y 3 Ceibal (18 en total, las de mayor uso), *downlink* y *uplink*.
- Histograma de tráfico por aplicación: 3 bins.
- Histograma de días por aplicación: 3 bins.
- Total de características por dirección `MAC` (por dispositivo): $20 \times 2 \times 3 + 18 \times 2 \times 2 \times 3 = 336$

Se cuenta entonces con un total de 336 características por dispositivo, las cuales están compuestas por la distribución del uso de la red según su consumo de ancho de banda y su utilización, construidos a partir de histogramas.

3.3. Conjunto de datos

Se cuenta con datos correspondientes a 8 semanas, desde la semana 19 de 2017 (segunda semana de mayo) y la 26 (última semana de junio). Para este trabajo, se consideró solamente los datos de las primeras 4 semanas. El conjunto de datos proporcionado no cuenta con los patrones de menor actividad (fueron filtrados según el siguiente criterio: al menos 5 días de actividad, tráfico acumulado mayor o igual a 1 MB).

De esta forma, el conjunto de datos queda compuesto por un total de 46397 patrones con 336 características, distribuidos en 3 clases, según la siguiente relación:

- `ceibal-laptop`: 11442 (24,66 %)
- `ceibal-tablet`: 4160 (8,97 %)
- `no-ceibal`: 30795 (66,37 %)

A partir de una primera observación, se puede ver que existe un desbalance importante entre la cantidad de patrones disponible para cada clase. Este problema se abordará en la Sección 5.

4. Ganando intuición: primera clasificación

Para ganar intuición sobre el problema, se realizó una primera clasificación utilizando el conjunto de datos crudo. En primera instancia, se separó el conjunto de datos en 80 % entrenamiento y 20 % test. Dado que se cuenta con desbalance de clases, el sorteo se realizó de forma supervisada de forma de mantener las proporciones de los datos de cada clase en cada subconjunto. A continuación, se normalizó el conjunto de entrenamiento utilizando un normalizador del tipo min/max, es decir, los datos de cada feature son escalados en el intervalo $[0 \dots 1]$ según la siguiente relación:

$$f_{norm} = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{1}$$

La estadística calculada sobre el conjunto de entrenamiento (en este caso, valores mínimos y máximos para cada feature) fue utilizada para escalar el conjunto de test. Este es el procedimiento habitual, donde se busca que el conjunto de test permanezca aislado y es solamente utilizado al momento de validar. Todas las estadísticas se calculan sobre el conjunto de entrenamiento y luego

son aplicadas al conjunto de test. De esta forma se evita sesgar al estimador.

Para finalizar, se entrenaron dos clasificadores: regresión logística multinomial (también llamado *softmax*) y random forest. Para este último, se realizó una búsqueda de hiperparámetros en una grilla, de forma de obtener el subconjunto que lograra los mejores resultados.

En la Tabla 1 se muestran los resultados de las métricas obtenidas a partir de la clasificación. Se puede ver que en promedio, los valores de exactitud sobre el conjunto de test son muy altos (por encima del 92 % en ambos clasificadores). Sin embargo, al observar los valores de recall, precisión y F1-score por cada clase, se puede ver que el resultado para la clase `ceibal-tablet` es sensiblemente menor que para el resto, siendo más evidente en las métricas recall y F1-score. Dado que se trata de un problema multi-clase, las métricas mostradas corresponden a un enfoque del tipo "*one-vs-all*", donde se compara cada clase frente al resto, como si se tratase de un problema binario. En dicho contexto, una muestra "positiva" refiere a una muestra correspondiente a la clase en cuestión, mientras que una muestra "negativa" corresponde a una muestra perteneciente a cualquiera de las clases restantes. Bajo este esquema, el recall se calcula como la proporción de muestras clasificadas de forma correcta como positivas, sobre el total de muestras positivas:

$$\text{recall} = \frac{TP}{TP + FN} \tag{2}$$

donde TP representa a las muestras clasificadas correctamente como positivas (*true positives*) y FN representa a las muestras de clasificadas incorrectamente como negativas (*false negatives*). Un valor bajo en recall implica que el clasificador tiene dificultad para poder distinguir las muestras de esta clase respecto del resto. De forma más general, se puede decir que el recall calcula la exactitud por clase: dada una clase 'A', corresponde a la proporción de muestras de la clase 'A' clasificadas de forma correcta, sobre el total de muestras de la clase 'A' (filas en la matriz de confusión).

Por otro lado, la precisión mide la habilidad del clasificador de no etiquetar muestras correspondientes a otras clases de forma incorrecta:

$$\text{precision} = \frac{TP}{TP + FP} \tag{3}$$

donde FP representa a las muestras incorrectamente clasificadas como positivas (*false positives*). De forma más general, dada una clase 'A', la precisión se calcula como la cantidad de muestras de la clase 'A' clasificadas de forma correcta, sobre el total de muestras clasificadas como 'A' (columnas en la matriz de confusión).

Finalmente, el F1-Score es la media armónica entre recall y precisión, por lo que representa una forma de combinar ambas métricas:

$$\text{F1-score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \tag{4}$$

Métrica	Softmax			Random forest		
	`laptop-ceibal`	`tablet-ceibal`	`no-ceibal`	`laptop-ceibal`	`tablet-ceibal`	`no-ceibal`
Accuracy (train)		0.9289			0.9995	
Accuracy (test)		0.9278			0.9471	
G-mean		0.8073			0.8692	
Precision	0.9553	0.8726	0.9237	0.9547	0.8818	0.9514
Recall	0.8869	0.6010	0.9872	0.9292	0.7176	0.9847
F1-score	0.9198	0.7117	0.9544	0.9418	0.7913	0.9678

Tabla 1: Métricas obtenidas en la primera clasificación realizada a partir del conjunto de datos.

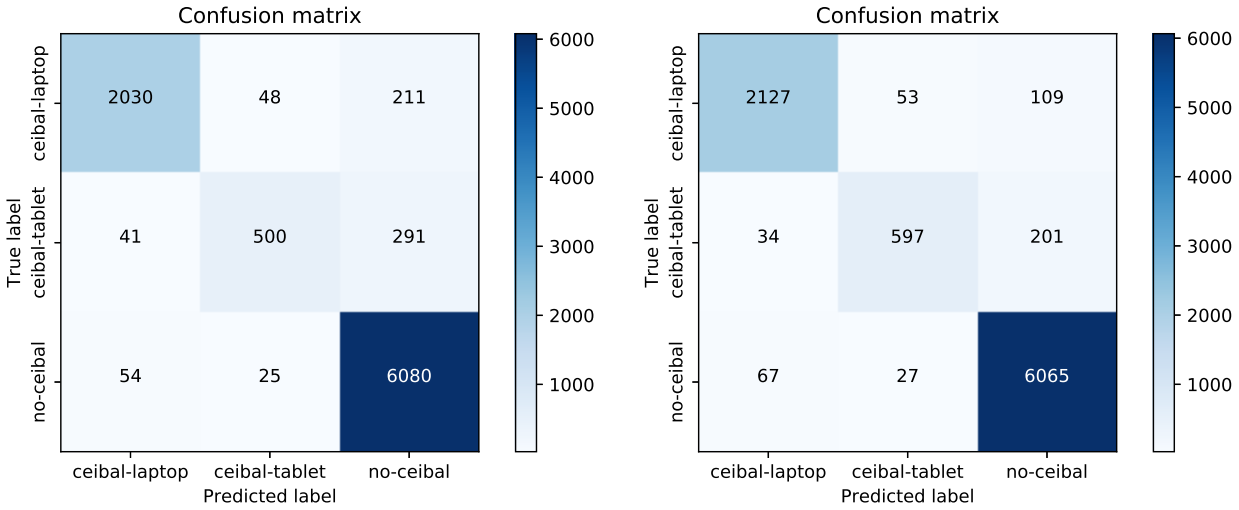
De forma complementaria, en la Figura 1 se muestran las matrices de confusión para cada clasificador. Aquí se puede ver que la mayor cantidad de confusiones se dan entre la clase `ceibal-tablet` y `no-ceibal` (el clasificador tiene dificultades para diferenciar entre estas clases), lo cual se refleja en el bajo valor de recall obtenido en la clasificación. Se puede ver que el 24,2 % (random forest) de las muestras del total de tablets son confundidas con dispositivos personales.

En la Tabla 1 también se incluyó la media geométrica de las exactitudes asociadas a cada clase (`G-mean`) la cual es una medida adecuada para ser utilizada como medida de desempeño en escenarios donde existe desbalance de clases [1]:

$$\text{G-mean} = \left(\prod_{i=1}^{i=n} \frac{TP_i}{TP_i + FN_i} \right)^{1/n} \tag{5}$$

donde n corresponde a la cantidad de clases. Esta medida es la que se intentará mejorar en el proceso de clasificación.

Por último, también se destaca que el desempeño obtenido entre ambos clasificadores fue muy similar, obteniendo mejores resultados con random forest. También se puede ver que en ninguno de los casos existe sobreajuste, ya que la distancia entre los valores de exactitud entre train y test es pequeña. De ahora en más, y para el resto de los análisis presentados en este informe, el clasificador a utilizar será el random forest.



(a) Softmax.

(b) Random forest.

Figura 1: Matrices de confusión obtenidas de la primera clasificación a partir del conjunto de datos. Observar que en ambos clasificadores la mayor cantidad de confusiones se dan al diferenciar tablets Ceibal respecto a dispositivos de uso personal.

4.1. Visualización

De modo de seguir ganando intuición sobre el problema, se decidió realizar extracción de características utilizando PCA de 2 componentes, con el fin de poder visualizar al conjunto de datos. En la Figura 2 se muestra el gráfico de los vectores de componentes del PCA. El patrón discreto presente en los datos puede explicarse debido a que están compuestos por frecuencias a partir de histogramas, como fue explicado en la Sección 3.2. También se puede ver que en esta representación, las clases están totalmente solapadas.

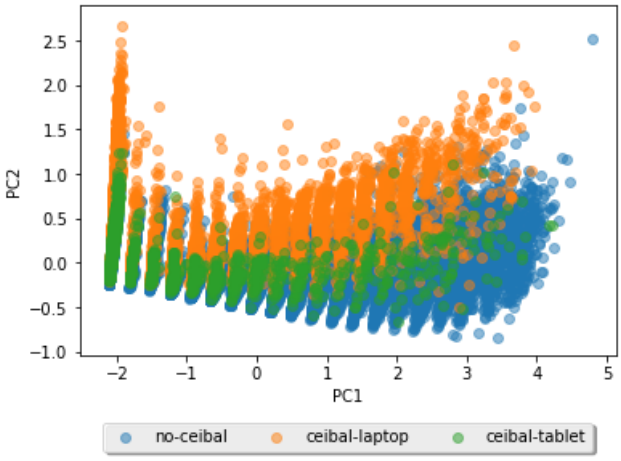


Figura 2: PCA de 2 componentes para visualizar el conjunto de datos. Los patrones pueden explicarse debido a que el conjunto de datos toma valores discretos, al tratarse de frecuencias obtenidas a partir de histogramas.

4.2. Selección y extracción de características

Un resultado a destacar a partir del PCA, es que el total de varianza acumulada de los datos teniendo en cuenta solamente las 2 primeras componentes es del 89 %. A partir de esto, se realizó una descomposición en una mayor cantidad de componentes y se observó que con las primeras 4 se obtiene el 93, 6 %. Esto motivó a realizar un proceso de selección y extracción de características como paso previo a la clasificación, para estudiar si existen features que por sí solos —o combinados— tienen mayor poder de discriminación que el conjunto completo.

A continuación se presentan los resultados de realizar el proceso de selección y extracción de manera individual y luego combinados. En todos los casos se utilizó un `pipeline` de forma que cada paso se ejecute de forma secuencial, utilizando una búsqueda de mejores hiperparámetros en una grilla.

4.2.1. Solo selección (chi-cuadrado)

El método de selección de características elegido fue chi-cuadrado (χ^2), el cual es utilizado para probar la independencia entre cada feature y la clase. El método requiere como entrada que se especifique la cantidad de características a elegir. Se buscó seleccionar el mejor conjunto de entre 100, 150, 200 y 250 características, obteniéndose los mejores resultados al elegir las "mejores" 250 (mejores desde el punto de vista del poder de discriminación).

En la Tabla 2 y en la Figura 3 se muestran las métricas y la matriz de confusión, respectivamente, obtenidas en este caso. Se puede ver que se logra una leve mejora en la clasificación para las clases `laptop-ceibal` y `no-ceibal`, mientras que para la clase `tablet-ceibal` los resultados son muy similares al caso sin preprocesar.

Métrica	χ^2 + Random forest		
	`laptop-ceibal`	`tablet-ceibal`	`no-ceibal`
Accuracy (train)		0.9927	
Accuracy (test)		0.9493	
G-mean		0.8706	
Precision	0.9600	0.8989	0.9507
Recall	0.9336	0.7164	0.9865
F1-score	0.9466	0.7973	0.9683

Tabla 2: Resultados de la clasificación utilizando selección de características mediante χ^2.

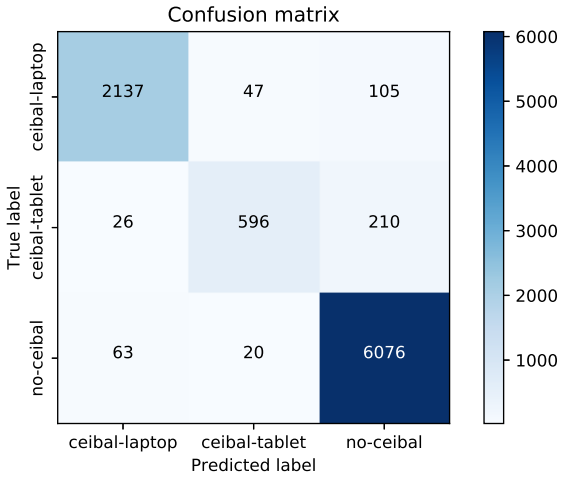


Figura 3: Matriz de confusión.

4.2.2. Solo extracción (PCA)

Como método de extracción de características se eligió PCA, el cual proyecta los datos en la dirección ortogonal que contiene mayor varianza. De forma similar a χ^2, PCA requiere que

se especifique como entrada la cantidad de componentes a utilizar. Se buscó encontrar la mejor proyección utilizando 4, 10, 20 y 50 componentes, siendo 50 el que obtuvo mejores resultados respecto a la clasificación. En la Tabla 3 y en la Figura 4 se muestran los resultados y la matriz de confusión, respectivamente. En este caso, los resultados son levemente inferiores respecto al caso de utilizar selección y respecto al caso original (con el conjunto total de características). PCA es una técnica que funciona de forma no supervisada (no tiene en cuenta información sobre las clases) y asume que la varianza del conjunto de datos es una medida que tiene poder de discriminabilidad. Dado los resultados, es posible que para este problema en particular, PCA proyecte los datos en direcciones que no facilitan la clasificación.

Métrica	PCA + Random forest		
	`laptop-ceibal`	`tablet-ceibal`	`no-ceibal`
Accuracy (train)		0.9992	
Accuracy (test)		0.9400	
G-mean		0.8471	
Precision	0.9390	0.9005	0.9441
Recall	0.9153	0.6743	0.9851
F1-score	0.9270	0.7711	0.9642

Tabla 3: Resultados de la clasificación utilizando extracción de características mediante PCA.

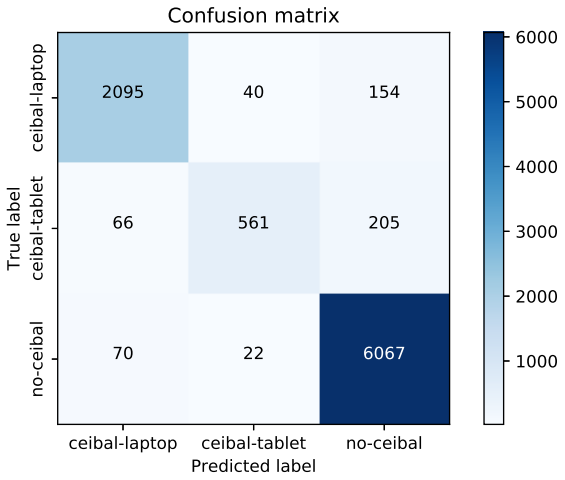


Figura 4: Matriz de confusión.

4.2.3. Combinando selección y extracción

Como forma complementaria, en esta sección se presentan los resultados de aplicar de forma secuencial el proceso de selección y el de extracción. Los mejores parámetros encontrados en este caso coinciden con los anteriores: chi-cuadrado seleccionó las mejores 250 características que luego PCA las combinó para encontrar las "mejores" 50 componentes. Los resultados se presentan en la Tabla 4 y en la Figura 5. Se puede ver que, si bien los resultados son similares al caso en que se utilizaron todas las características, la mayoría de las métricas desmejoraron respecto al caso original.

Métrica	χ^2 + PCA + Random forest		
	`laptop-ceibal`	`tablet-ceibal`	`no-ceibal`
Accuracy (train)		0.9991	
Accuracy (test)		0.9404	
G-mean		0.8540	
Precision	0.9447	0.8879	0.9443
Recall	0.9104	0.6947	0.9847
F1-score	0.9273	0.7795	0.9641

Tabla 4: Resultados de la clasificación utilizando selección (χ^2) y extracción (PCA) de características.

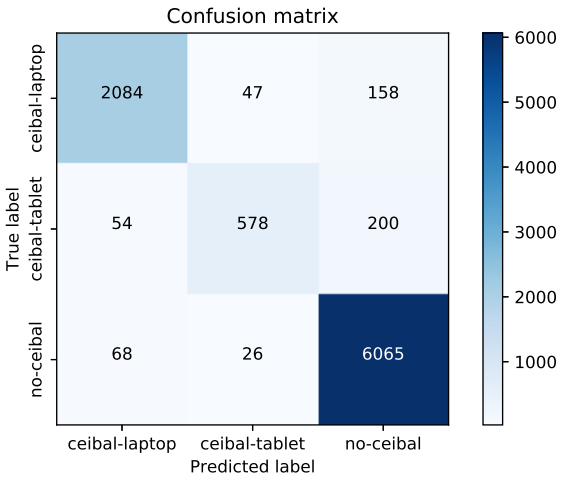


Figura 5: Matriz de confusión.

5. Desbalance de clases

En esta Sección se estudia el problema de desbalance de clases mencionado en la Sección 3.3. En particular, se observó que el 24,2 % de las muestras de tipo `ceibal-tablet` son confundidas con dispositivos `no-ceibal` (Figura 1b). Dos posibles interpretaciones para este problema pueden ser, por un lado, la posibilidad de que el propio desbalance de clases esté afectando al desempeño del clasificador; o por otro, puede ser razonable que el clasificador tenga problemas para distinguir entre estos tipos de dispositivos, dado que en su mayoría, los `no-ceibal` están conformados por dispositivos móviles (celulares y tablets) los cuales pueden tener un comportamiento similar respecto al uso que `ceibal-tablet`.

A continuación se aborda el problema del desbalance con el objetivo de mejorar el desempeño del clasificador y poder inferir si el menor desempeño en la clasificación de la clase `ceibal-tablet` se debe a este fenómeno. La estrategia a tomar se basa en submuestrear las clases mayoritarias (`ceibal-laptop` y `no-ceibal`) para llevarlas al tamaño de la clase minoritaria (`ceibal-tablet`).

El clasificador a utilizar en todos los casos es el random forest, con un paso previo de selección de características utilizando chi-cuadrado, dado que este fue el que brindó mejores resultados.

5.1. Submuestreo aleatorio

La primera forma para balancear al conjunto de datos (tal vez la más directa) es la de realizar un submuestreo aleatorio de los patrones de las clases mayoritarias, para llevarlas al tamaño de la clase minoritaria. Dado que se cuenta con 4160 patrones de la clase `ceibal-tablet`, el conjunto total de datos pasa a estar conformado por un total de 12480 patrones (4160 por clase).

Una vez conformado el nuevo conjunto, se procedió a dividirlo en 80 % train y 20 % test. Luego, se normalizaron los datos aplicando min/max sobre el conjunto de entrenamiento, para posteriormente aplicar dicho escalamiento sobre el conjunto de test.

Para finalizar, se utilizó un `pipeline` para incluir los pasos de selección y de clasificación. Al igual que en la primera clasificación, se realizó una búsqueda de hiperparámetros en una grilla. Los mejores hiperparámetros (los que maximizan el valor del `G-mean`) son los siguientes:

- Chi-cuadrado
  - Cantidad de características seleccionadas (`k`): 200
- Random forest
  - Medida de impureza (`criterion`): gini
  - Número de estimadores (`n_estimators`): 70
  - Máxima profundidad del árbol (`max_depth`): 20
  - Mínima cantidad de muestras por hoja (`min_samples_leaf`): 1

En la Tabla 5 se muestra el resultado de la clasificación y en la Figura 6 la matriz de confusión. Se presenta el promedio de 5 sorteos aleatorios distintos, de forma de evitar que los resultados estén sesgados a un sorteo particular. Se puede ver una mejora en las métricas asociadas a la clase `ceibal-tablet` respecto a la clasificación con el conjunto desbalanceado. Observar que el valor de recall para esta clase aumentó un 10,7 % (71,6 % a 82,3), lo que implica una disminución en la cantidad de falsos negativos. A su vez, el valor de la media geométrica mejoró un 2,6 % (87,1 % a 89,7 %).

Si bien hay una mejora respecto a las métricas obtenidas con el conjunto de datos desbalanceado, aún persiste un grupo no despreciable de patrones que el clasificador no puede distinguir entre las clases `ceibal-tablet` y `no-ceibal`. Esto sugiere que, si bien el desbalance es un problema, la gran proporción de dispositivos móviles en la clase `no-ceibal` dificulta la discriminación entre estas dos clases.

Métrica	Random forest		
	`laptop-ceibal`	`tablet-ceibal`	`no-ceibal`
Accuracy (train)		0.9693	
Accuracy (test)		0.8983	
G-mean		0.8966	
Precision	0.9639	0.9022	0.8377
Recall	0.9418	0.8228	0.9303
F1-score	0.9527	0.8606	0.8815

Tabla 5: Resultados de la clasificación luego de realizar submuestreo aleatorio de las clases mayoritarias.

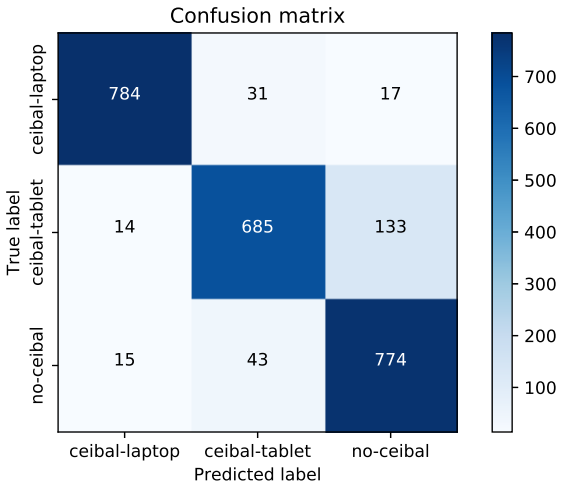


Figura 6: Matriz de confusión.

5.2. Submuestreo utilizando K-means

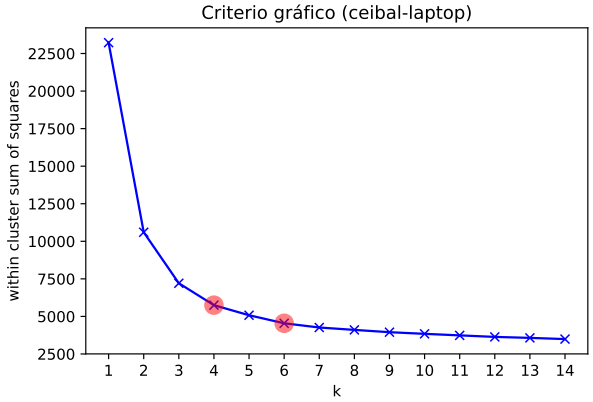
En esta Sección se presenta una alternativa al submuestreo aleatorio presentado anteriormente. La idea es realizar clustering no supervisado sobre los conjuntos de patrones correspondientes a las

clases mayoritarias de forma de segmentarlos en clusters, o grupos. Luego, realizar el submuestreo a partir de muestras aleatorias, pero en esta oportunidad, elegidas de forma equitativa sobre cada uno de los grupos inferidos.

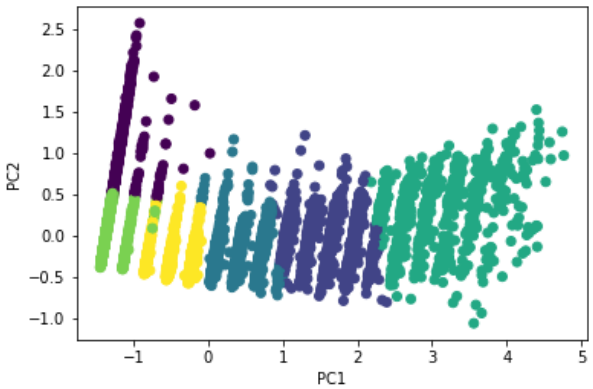
El algoritmo de clustering elegido es K-means. El procedimiento para la confección del conjunto de entrenamiento se describe a continuación:

1. Se conforma el conjunto de test sorteando de forma supervisada una cantidad de muestras igual al 20 % del tamaño de la clase minoritaria (`ceibal-tablet`, 832 muestras). De esta forma, se llega a un conjunto de test de tamaño 2496 muestras $\Rightarrow$ (X_{test}, y_{test})
2. El resto de las muestras es utilizado como conjunto de train auxiliar $\Rightarrow$ (X'_{train}, y'_{train})
3. Normalizo (min/max) utilizando X'_{train} y aplico dicha normalización a X_{test}.
4. Divido X'_{train} según cada etiqueta.
5. Para las etiquetas mayoritarias (`ceibal-laptop` y `no-ceibal`):
   - Aplico K-means y sampleo de forma equitativa una cantidad de muestras igual al 80 % del tamaño de la clase `ceibal-tablet` (4160 muestras) $\Rightarrow$ $X_{train}[\texttt{ceibal-laptop}, \texttt{no-ceibal}]$
6. Genero el conjunto de entrenamiento de tamaño 12480 muestras, a partir de $X_{train}[\texttt{ceibal-laptop}]$, $X_{train}[\texttt{no-ceibal}]$ y $X'_{train}[\texttt{ceibal-tablet}]$ $\Rightarrow$ (X_{train}, y_{train})

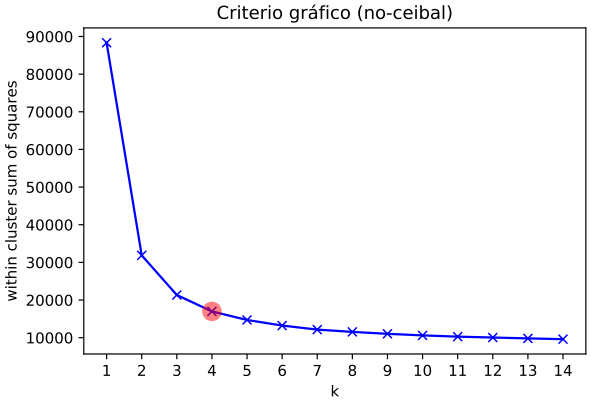
La elección del número de clusters para cada conjunto se realizó mediante el criterio gráfico (también llamado "método del codo"—*elbow method*). La idea consiste en elegir una cantidad de clusters de forma tal que agregar más clusters no aporte información. Para cada número de clusters entre 1 y K, se minimiza de forma iterativa la función de criterio (la suma de los errores al cuadrado, *sse*). El objetivo es elegir el número de clusters k^* en el cual la función de criterio baje poco con respecto a $k^* - 1$. En la Figura 7 se muestra el resultado del criterio gráfico para cada una de las clases mayoritarias. Se destaca en color rojo los valores candidatos. Para el caso de la clase `ceibal-laptop` se eligieron 6 clusters, mientras que para la clase `no-ceibal` se eligieron 4. También se muestra la distribución de patrones según cada cluster mediante un PCA de 2 componentes.



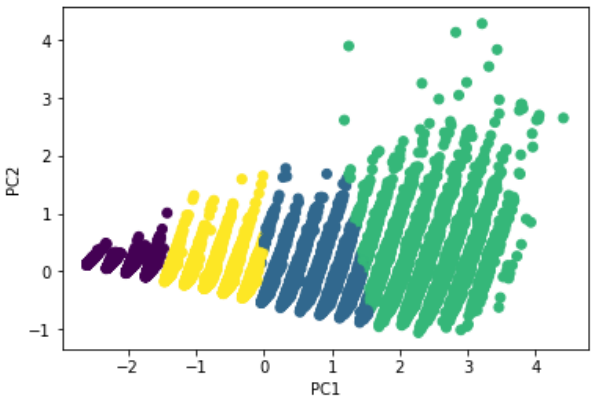
(a) Método gráfico para la elección de la cantidad de clusters. En rojo se destacan los valores candidatos (clase `ceibal-laptop`).



(b) PCA de 2 componentes para la clase `ceibal-laptop`. Se distingue en diferentes colores el resultado del K-means (6 clusters).



(c) Método gráfico para la elección de la cantidad de clusters. En rojo se destaca el valor candidato (clase `no-ceibal`).



(d) PCA de 2 componentes para la clase `no-ceibal`. Se distingue en diferentes colores el resultado del K-means (4 clusters).

Figura 7: Elección de la cantidad de clusters según el criterio gráfico y PCA de 2 componentes sobre los patrones de cada clase mostrando el resultado del K-means.

En la Tabla 6 y en la Figura 8 se muestran los resultados de la clasificación utilizando este método para el submuestreo. En este caso también se presenta el promedio de los resultados de 5 sorteos. Se puede ver que si bien los resultados son similares, no existe una mejora respecto al caso del muestreo aleatorio.

Métrica	Random forest		
	`laptop-ceibal`	`tablet-ceibal`	`no-ceibal`
Accuracy (train)		0.9996	
Accuracy (test)		0.8839	
G-mean		0.8825	
Precision	0.9686	0.8672	0.8252
Recall	0.9240	0.8173	0.9103
F1-score	0.9457	0.8415	0.8656

Tabla 6: Resultados de la clasificación luego de realizar submuestreo utilizando K-means de las clases mayoritarias.

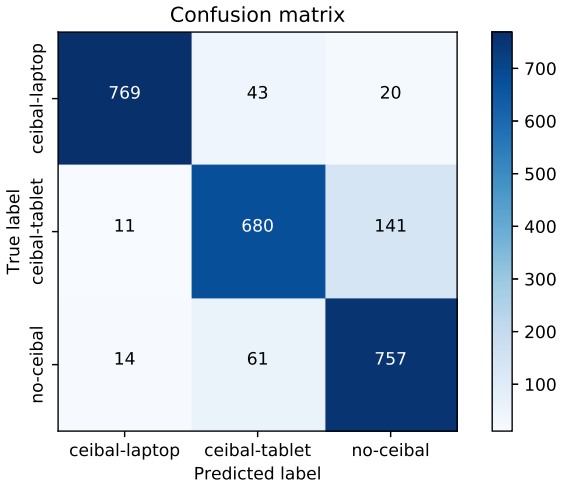


Figura 8: Matriz de confusión.

6. Comparación de resultados

En esta Sección se realiza la comparación formal de cada uno de los resultados presentados a lo largo de este informe. Como fue explicado en la Sección 4, la medida a comparar es el `G-mean`. Para realizar la comparación entre cada resultado, se busca encontrar los intervalos de confianza asociados a cada medida. A partir de la ecuación 5, se puede calcular la derivada de la media

geométrica respecto a cada uno de las exactitudes por etiqueta (observar que coinciden con el valor de recall, ver ecuación 2). Se puede ver que dicha derivada depende del propio valor de la media geométrica. Siendo g el valor de la media geométrica y r el valor del recall, la derivada de g respecto de r resulta:

$$\begin{aligned}\frac{\partial g}{\partial r_i} &= \frac{1}{n}\left(\prod_{i=1}^{i=n} r_i\right)^{\frac{1}{n}-1} \prod_{j\neq i}^{j=n} r_j \\ &= \frac{1}{n}\frac{\left(\prod_{i=1}^{i=n} r_i\right)^{1/n}}{\prod_{i=1}^{i=n} r_i} \prod_{j\neq i}^{j=n} r_j \\ \Rightarrow \quad \frac{\partial g}{\partial r_i} &= \frac{1}{n}\frac{g}{r_i} \end{aligned} \tag{6}$$

Por otro lado, la incertidumbre del error en el recall puede ser calculada mediante la siguiente expresión [2]:

$$\mu_r = k\sqrt{\frac{e_r\,(1-e_r)}{n}} \tag{7}$$

donde la constante k es la que regula el nivel de confianza con el que se quiere estimar el error (por ejemplo, $k = 1{,}96$ implica que con un 95 % de probabilidad se está en dicho intervalo); n es la cantidad de muestras del conjunto de test y e_r es el error sobre el recall $(1 - recall)$.

Finalmente, a partir de las ecuaciones 6 y 7, y teniendo en cuenta la fórmula de propagación de incertidumbre, se llega a la siguiente expresión para el error sobre la media geométrica:

$$\mu_g = \frac{g}{n}\sqrt{\sum_{i=1}^{n}\left(\frac{\mu_{ri}}{r_i}\right)^2} \tag{8}$$

En la Tabla 7 se muestra el valor de las medias geométricas con su incertidumbre asociada (considerando $k = 1{,}96$ para un 95 % de probabilidad). Para complementar, en la Figura 9 se muestra de forma gráfica los valores centrales de media geométrica con su intervalo de confianza asociado. Se puede ver, por ejemplo, que los valores obtenidos con el conjunto de datos sin preprocesar y aplicando χ^2 como selección de características brindan prácticamente el mismo resultado. Por otro lado, si bien los casos de extracción mediante PCA y combinación de selección y extracción brindan valores centrales sensiblemente más bajos, existe cierto solapamiento entre los intervalos de confianza, lo que equivale a que, con cierta probabilidad (baja en el caso de solo PCA), ambos métodos deriven en resultados equivalentes. Finalmente, se puede ver que el mejor resultado se obtiene realizando submuestreo aleatorio (no existe solapamiento de intervalos con ninguno de los casos sin submuestrear). También es de destacar que el método de submuestreo mediante K-means es, con cierta probabilidad, equivalente al método de submuestreo aleatorio.

Método	G-mean (%)	μ_g (%)
Sin preproc.	86,9	1,3
Solo χ^2	87,1	1,3
Solo PCA	84,7	1,4
χ^2 + PCA	85,4	1,3
Sub. random	89,7	1,2
Sub. K-means	88,3	1,3

Tabla 7: Resultados de la media geométrica para cada caso estudiado y su incertidumbre asociada.

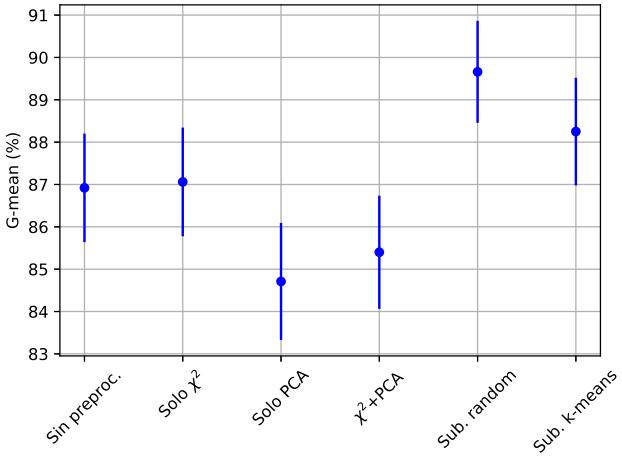


Figura 9: Gráfico de las medias geométricas con su intervalo de confianza asociado.

7. Conclusiones

En este trabajo se presenta un primer análisis sobre el poder de discriminación del vector de features definido con motivo de realizar segmentación de usuarios de las redes Wi-Fi de Plan Ceibal.

Estos primeros resultados permiten inferir que es posible diferenciar entre una laptop entregada por Ceibal del resto de los dispositivos, solamente a partir de datos de tráfico, con un alto nivel de confianza. Esto sugiere que el patrón de tráfico de los beneficiarios que hacen uso de laptopts Ceibal es distinto del de otro tipo de dispositivos.

También se observó que la gran cantidad de dispositivos móviles presentes en la categoría `no-ceibal` distorsiona los resultados del clasificador para la clase `ceibal-tablet`. Una posible solución a este problema sería separar en dos categorías a dicha clase, entre celulares y tablets y laptops.

Realizando selección de características se pudo ver que se mejora levemente el desempeño del clasificador, eligiendo entre 200 y 250 características de un total de 336. Además, se pudo ver que si bien aplicando extracción de características mediante PCA de 4 componentes se obtiene el 94 % de la varianza, esta características no es representativa del poder de discriminación para este problema particular.

Para sortear el problema de desbalance de muestras, se utilizaron 2 técnicas: submuestreo aleatorio y submuestreo mediante clustering no supervisado. Ambos métodos presentaron resultados similares, siendo superior el desempeño obtenido con el submuestreo aleatorio. Este último presenta una mejora sustancial respecto a los casos desbalanceados.

8. Trabajo futuro

Este trabajo brinda una primera exploración sobre el poder de discriminación que presenta el vector de features para la caracterización de usuarios a partir de datos de tráfico. Como trabajo

futuro se propone:

- Generar distintos conjuntos de datos a partir de distintos tipos de agregación temporal (en este trabajo se utilizaron datos de 4 semanas). El objetivo de esto sería estudiar cuál es la agregación óptima para la clasificación, lo cual determina la estacionareidad más habitual en los patrones de comportamiento. Se propone analizar la agregación de datos en 2, 3, 5, 6 y 8 semanas.
- Estudiar el poder predictivo de los clasificadores con el fin de determinar, por ejemplo, si los usuarios siguen un patrón de comportamiento regular en el tiempo. Para esto, se podría entrenar un clasificador con los datos de las primeras 4 semanas y verificar con las siguientes 4.
- Diferenciar los dispositivos móviles personales (no entregados por Ceibal) como una clase aparte, con el objetivo de estudiar las confusiones obtenidas entre las clases `ceibal-tablet` y `no-ceibal`.

Referencias

[1] Fiori, Marcelo, Matías Di Martino y Alicia Fernández: *An optimal multiclass classifier design.* En *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, páginas 480–485. IEEE, 2016. `https://doi.org/10.1109/ICPR.2016.7899680`.

[2] Mitchell, Tom M.: *Machine Learning.* WCB McGraw-Hill, 1997.

A. Archivos entregados

Se entrega adjunto a este informe los siguientes archivos:

- `datos_proyecto.pkl`: archivo `pickle` que conforma el conjunto de datos a analizar.
- `proyecto_1subset_analisis.ipynb`: `notebook jupyter` que contiene el análisis completo y clasificación.