Data Science y Energía

Cómo (casi) gané un concurso de Big Data

Grupo de Usuarios de R de Madrid

2017.02.28

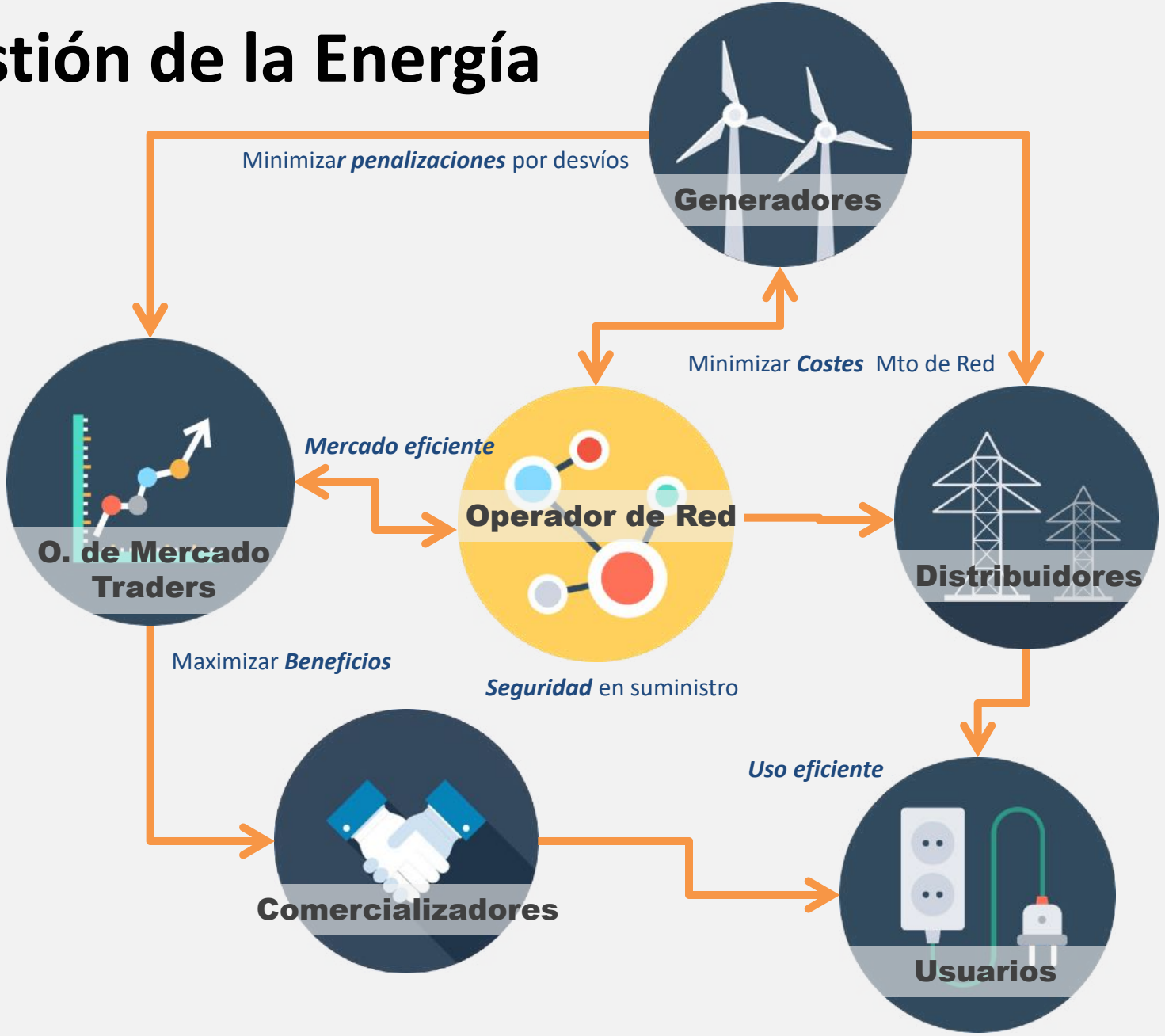
Paulino Tardáguila

@bicho_paulinho

paulino.tardaguila@meteologica.com

METEOLOGICA

Gestión de la Energía



Cambio de Escenario

- **Fuentes renovables** → incertidumbre
- **Generación distribuida** → Smart-Grids
- Implantación **contadores inteligentes**
- **Consumidor** demanda más **control e información**
- **Mantenimiento infraestructuras**

GRAN DISPONIBILIDAD DE DATOS → **NECESIDAD DE INFORMACIÓN**

- Generación
- Demanda
- Meteo
- Smart-meters
- IoT
- Precios (energía/materias primas)
- Incidencias

Modelos Estadísticos

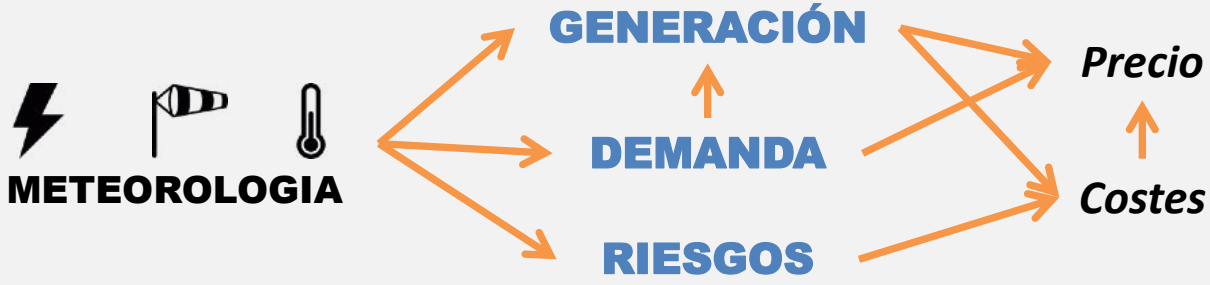
Cambio de Escenario

- **Fuentes renovables** → incertidumbre
- **Generación distribuida** → Smart-Grids
- Implantación **contadores inteligentes**
- **Consumidor** demanda más **control e información**
- **Mantenimiento infraestructuras**

GRAN DISPONIBILIDAD DE DATOS → **NECESIDAD DE INFORMACIÓN**

- Generación
- Demanda
- Meteo
- Smart-meters
- IoT
- Precios (energía/materias primas)
- Incidencias

Modelos Estadísticos

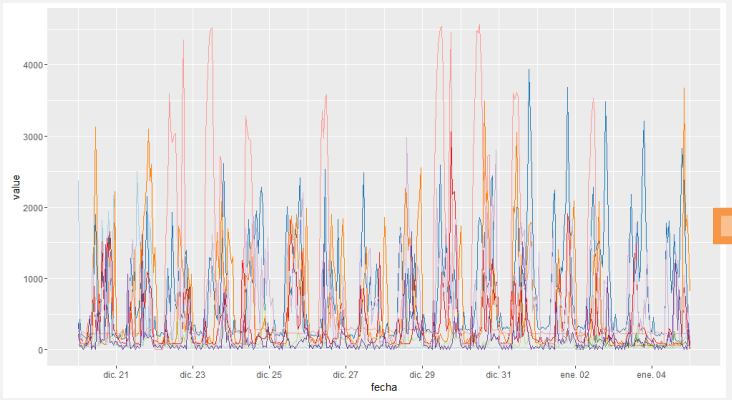


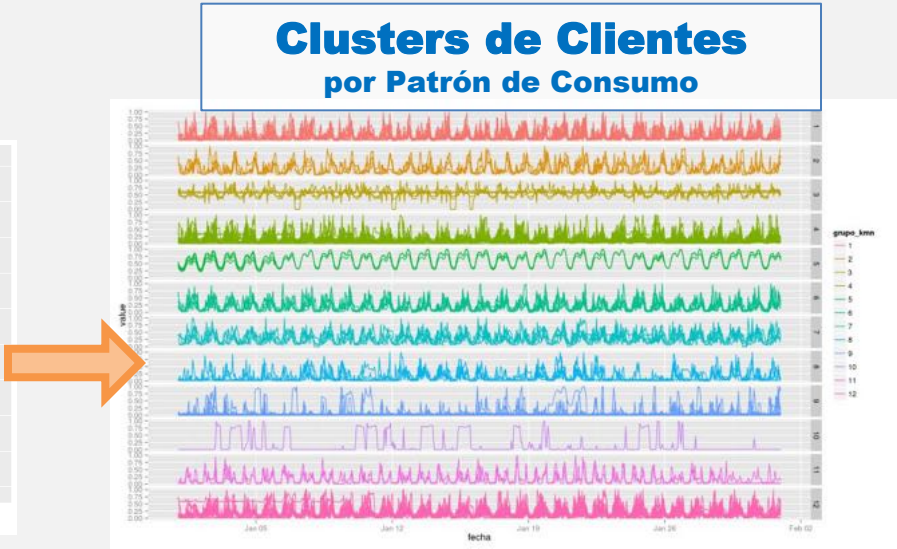
Endesa Datathon

- **Dataset**: consumo horario de 100k clientes + Info adicional
- **Objetivo**: *"Los participantes deberán proponer soluciones de valor añadido para los clientes, la compañía y la sociedad, ..."*
- **Criterios**:
  - Viabilidad
  - Valor para la comunidad
  - Innovación
  - Escalabilidad
- **Fases**:
  - Propuestas (40 preseleccionados)
  - Desarrollo (3 meses)
  - Evaluación (3 Finalistas)

La Idea

- ***Toda*** la información disponible→ Patrones
- ¿Cómo es el uso de la energía en toda la red de clientes?





Aplicaciones
Clusters Consumo

Distribuidora / Comercializadora: Predicción distribuida de Demanda

Operador de Red / Cliente Final: Sistema alternativo de tarificación (*clientes sin medidor inteligente*)

Viable → Sencillez: Métodos estadísticos + Computación

Escalable → Problema de tamaño real → Herramientas Big Data

Innovación + Valor Añadido

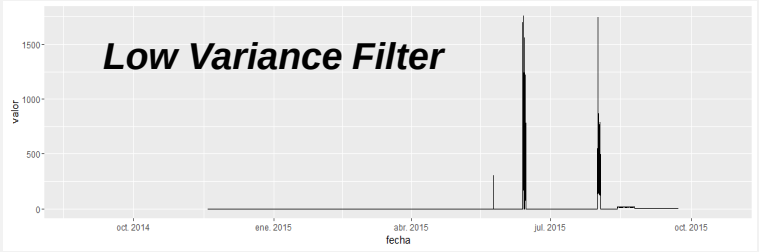
"Full Stack" Spark Shiny plotly ggplot2

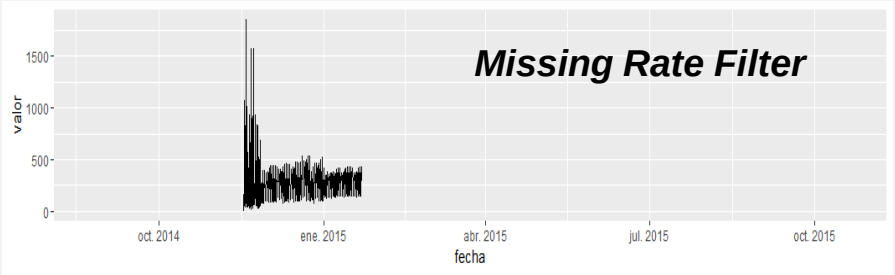
La Solución

Exploración & Limpieza de datos

Duplicados: Más de 1 valor por registro horario / ID cliente → sólo primera observación

Filtros Automáticos:





Periodos Missing: Relleno con valores *climatológicos. (*Pruebas con paquete Amelia)

Dataset de trabajo: >96500 series temporales de consumos horarios sincrónicos durante 1 año aprox. → Matriz ~ 96500 x 8500

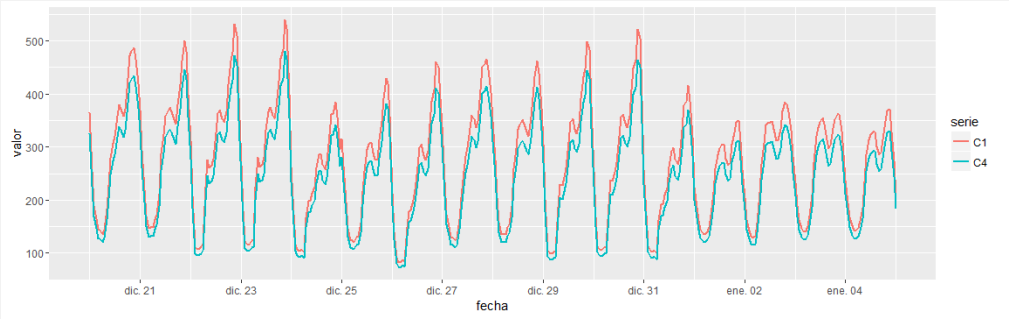
La Solución

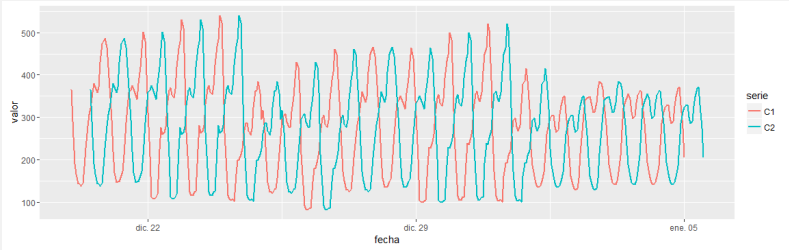
Metodología del Clustering

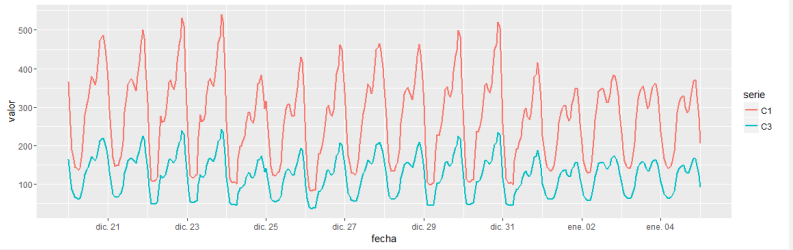
Objetivo: grupos con patrones de consumo similares

→ Distribución temporal: ***"forma" del perfil (Fase)***

→ Intensidad de uso: ***Valores similares (Nivel)***





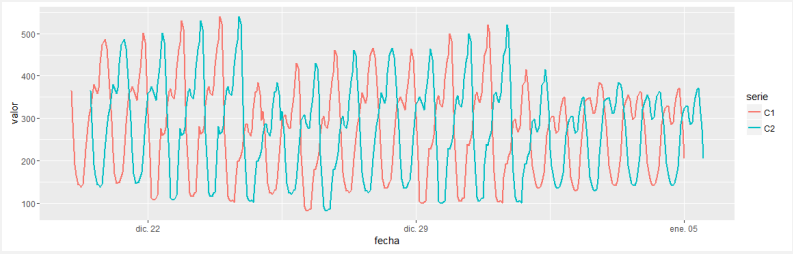


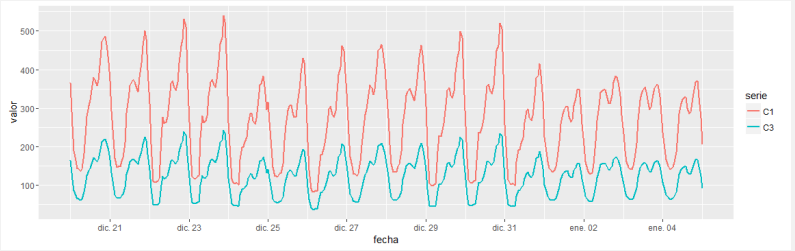
La Solución

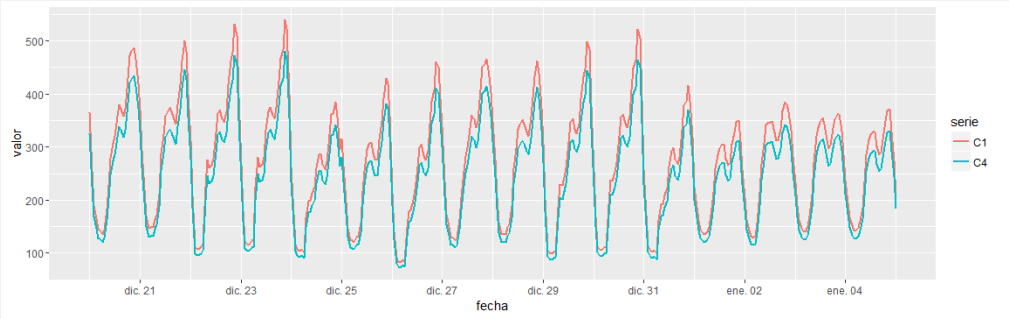
Metodología del Clustering

Objetivo: grupos con patrones de consumo similares

→ Distribución temporal: ***"forma" del perfil (Fase)***
→ Intensidad de uso: ***Valores similares (Nivel)***







TSClust: *"An R package for Time Series Clustering"*

- "*Indices de distancia adaptativos*": proximidad en nivel (valores) y fase (comportamiento):

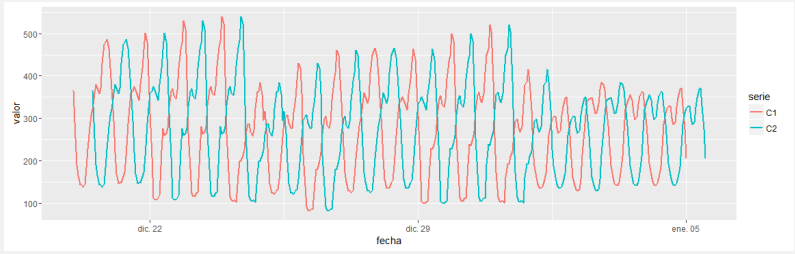
$$d_{CORT}\left(\boldsymbol{X}_T, \boldsymbol{Y}_T\right) = \phi_k\left[CORT\left(\boldsymbol{X}_T, \boldsymbol{Y}_T\right)\right] \cdot d\left(\boldsymbol{X}_T, \boldsymbol{Y}_T\right)$$

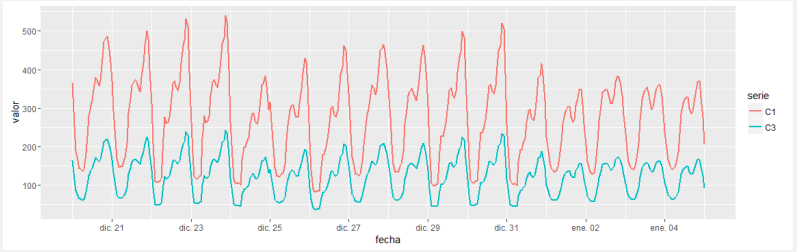
La Solución

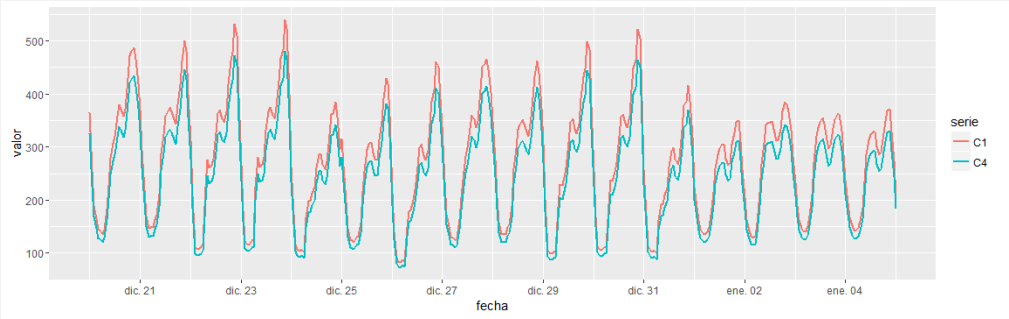
Metodología del Clustering

Objetivo: grupos con patrones de consumo similares

→ Distribución temporal: ***"forma" del perfil (Fase)***
→ Intensidad de uso: ***Valores similares (Nivel)***







TSClust: *"An R package for Time Series Clustering"*

- "*Indices de distancia adaptativos*": proximidad en nivel (valores) y fase (comportamiento):

$$d_{CORT}\left(\boldsymbol{X}_T, \boldsymbol{Y}_T\right) = \phi_k\left[CORT\left(\boldsymbol{X}_T, \boldsymbol{Y}_T\right)\right] \cdot d\left(\boldsymbol{X}_T, \boldsymbol{Y}_T\right)$$

- Medida de Similitud: ***Distancia Euclídea***
- Algoritmo: ***K-Means***

- ***Centroides*** → series representativas → Interpretabilidad
- ***Implementado*** en entornos de Big Data (Spark MLLIB)
- ***Resultados*** decentes

La Solución

Tecnología

Preparación de Datos

- `data.table`
- `dygraphs`
- `s3cmd`
- *Plataforma: Servidor local (32 cores 64 Gb) + Bucket S3*

Clustering

- Spark 1.4 (Hadoop 2.4)
- `pySpark` (`pyspark.mllib.clustering`)
- Plataforma: AWS EC2
  - Script `spark-ec2` (incl. En Spark)
  - Cluster: 1 master + 6 nodos (m3.xlarge 4 cores, 15Gb)

Resultados y Visualización

- `ggplot2, plotly, RColorBrewer, dygraphs`
- App web: `flexdashboard` (`rmarkdown` + `shiny`)
- Plataforma web: shinyapps.io

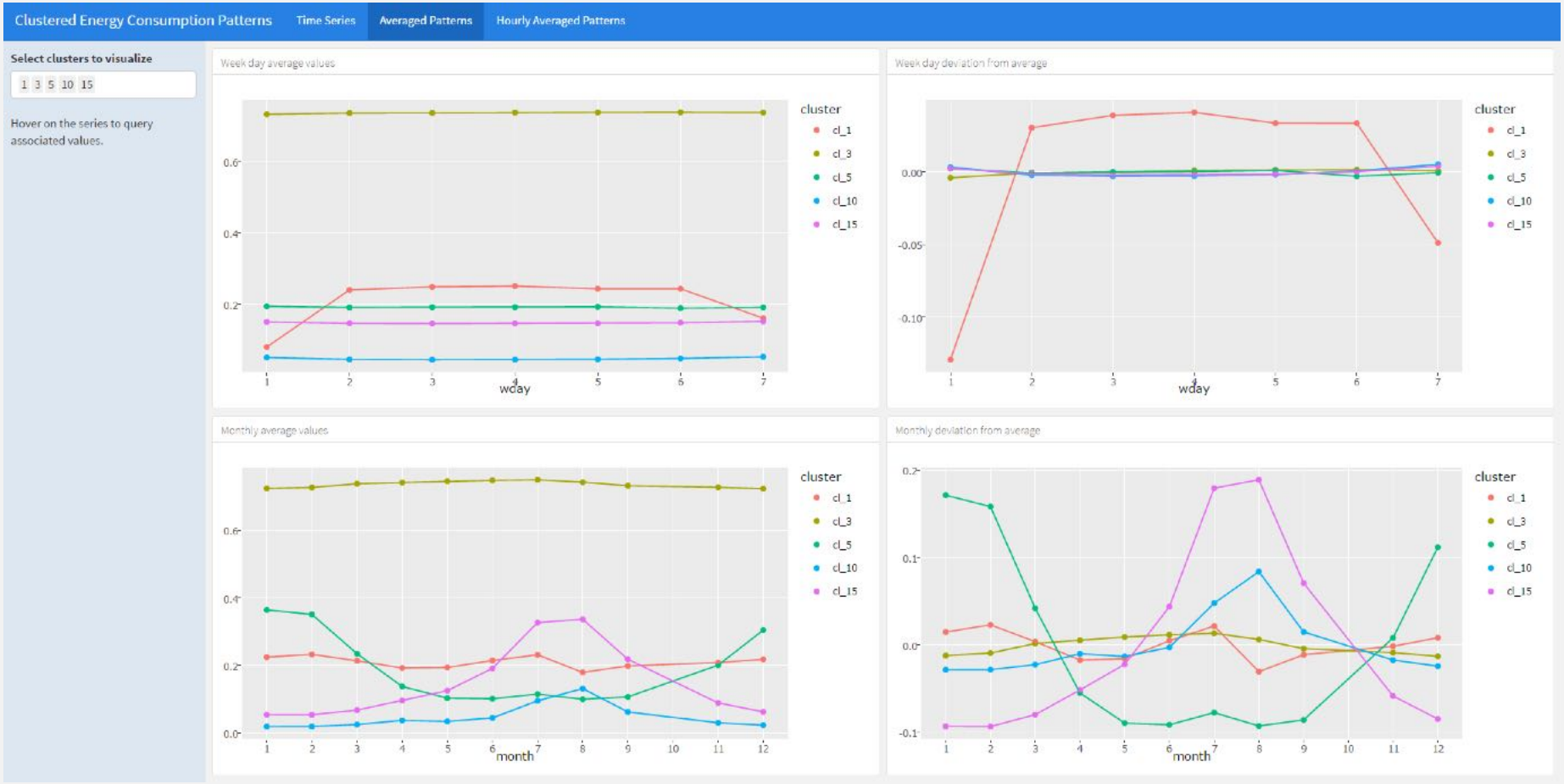
La Solución

Resultados

https://paulino-tardaguila.shinyapps.io/ClusterViz/

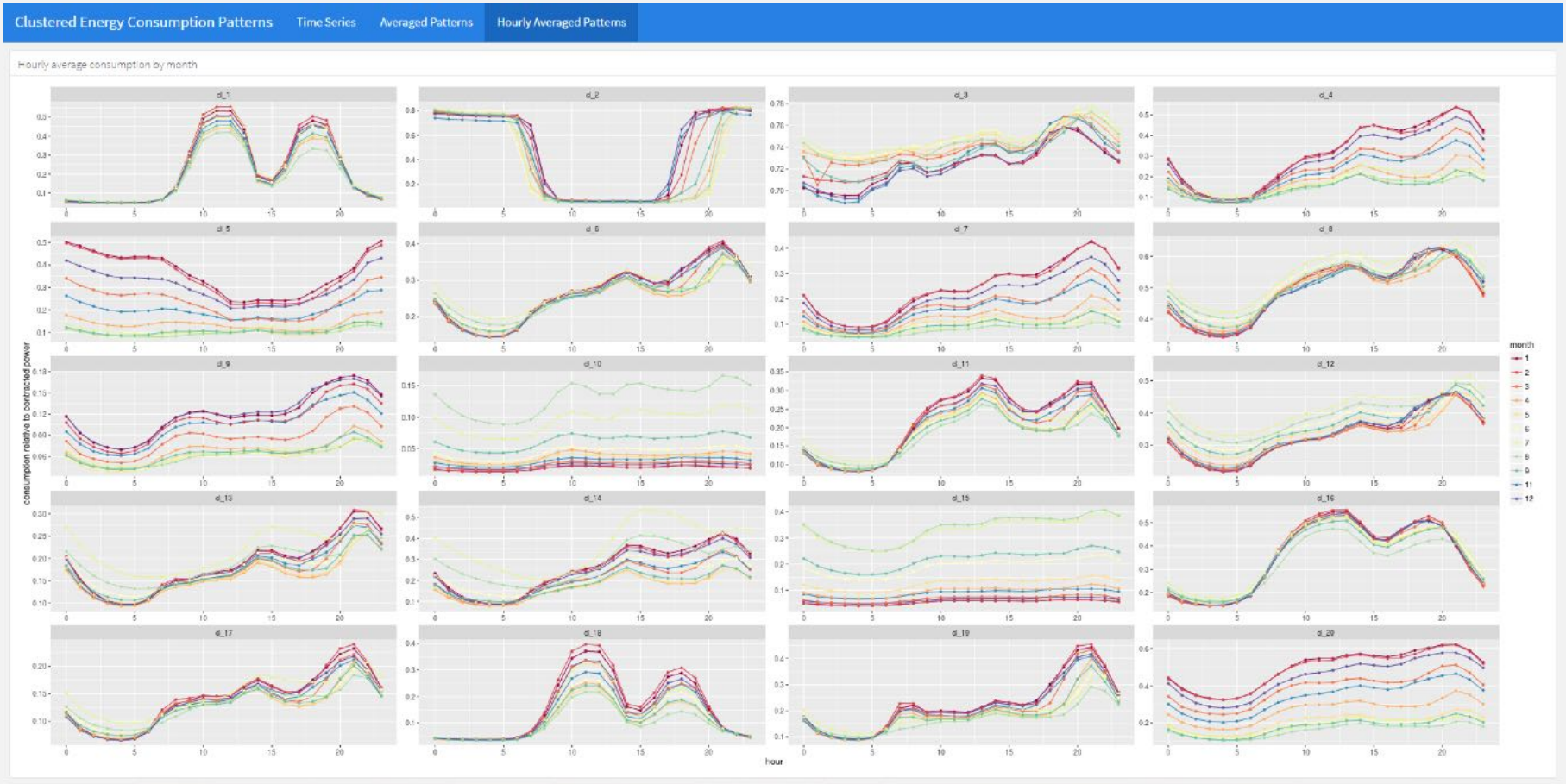
La Solución

Resultados



La Solución

Resultados



Aplicaciones

Predicción distribuida de demanda (Nodos Virtuales)

Aplicaciones

Predicción distribuida de demanda (Nodos Virtuales)

Agrupamiento a nivel Nodo

↓

Subnodos Virtuales

(por patron de demanda)

↓

Predicción por Subnodo

(Menor Varianza)

Mejor respuesta a predictores

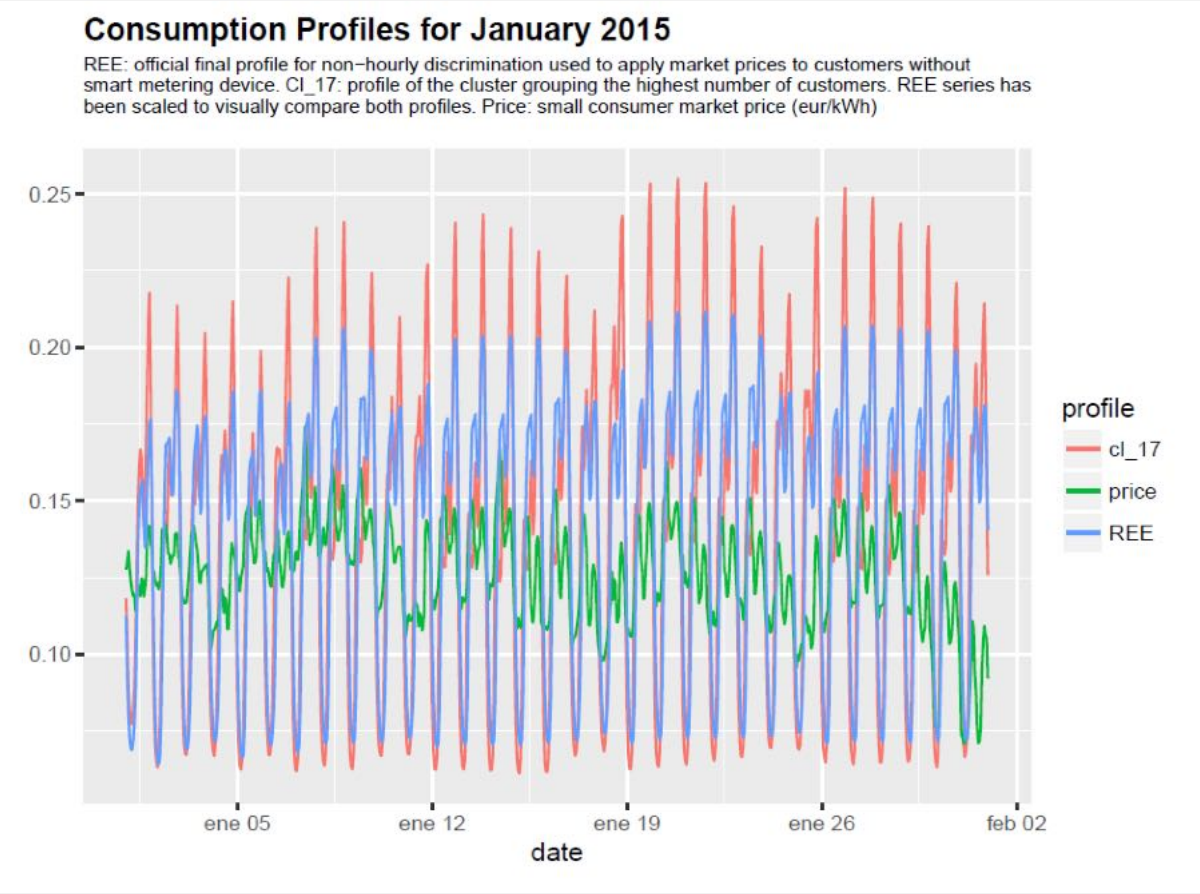
- *Meteo*
- *Calendario*

↓

Upscaling

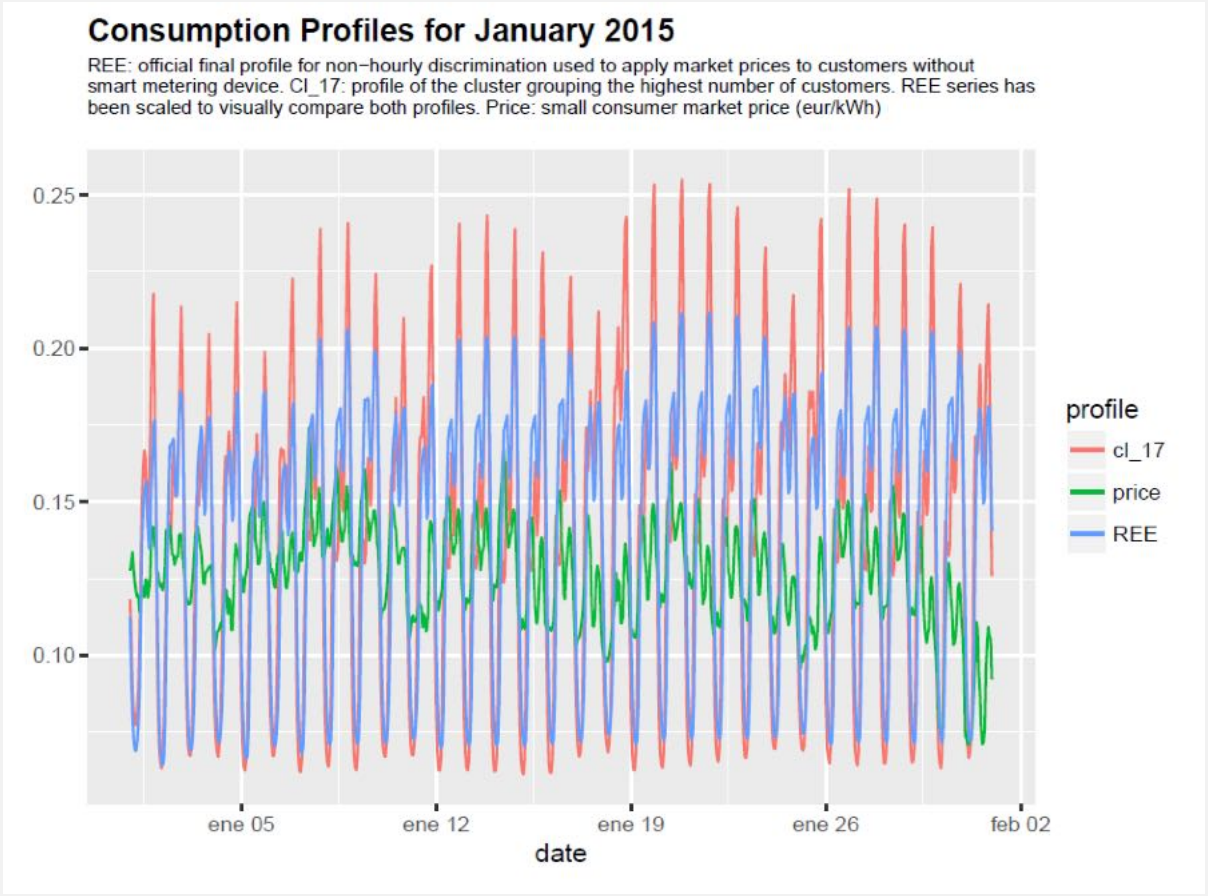
Aplicaciones

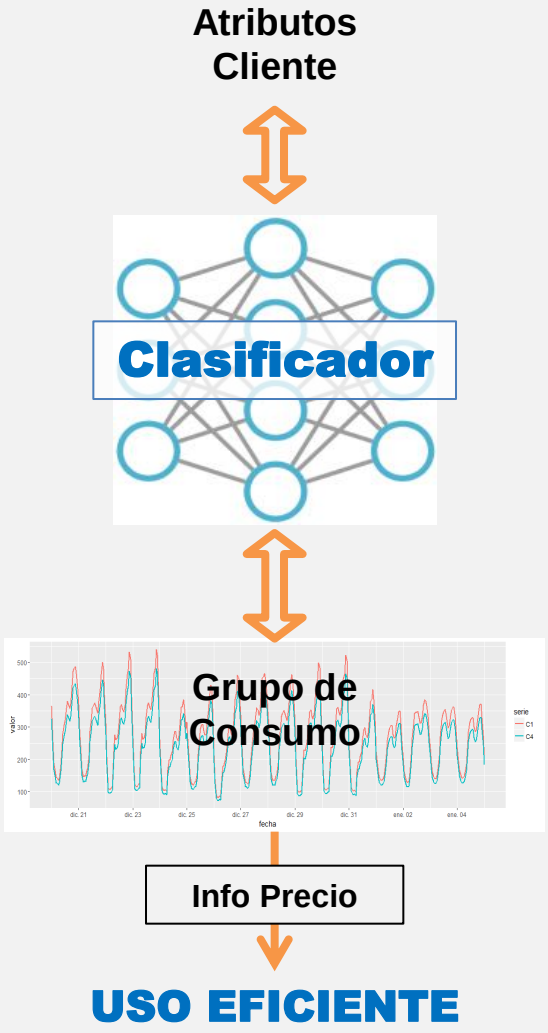
Perfiles Horarios mejorados para Tarificación Horaria



Aplicaciones

Perfiles Horarios mejorados para Tarificación Horaria





Reflexiones

Clave del Éxito → **APLICABILIDAD**

- ***Diseño*** simple y escalable (código y plataforma)
- ***Utilidad*** Directa → Problema real de la compra de energía
- ***Desarrollo*** Modular → Hitos parciales “entregables” + refinamiento posterior

Experiencia personal

- Integración de ***tecnología muy accesible*** (Spark, AWS)
- **R** → paraguas para ***todas las fases del proyecto:***
  - Tratamiento de la info
  - Big Data
  - Análisis de resultados
  - Reporting

¡Gracias por vuestra atención!