**Diplomatura en Ciencia de Datos, Aprendizaje Automático y sus Aplicaciones**

**Informe Trabajo Práctico 5 Mentoría**:

“Análisis del mercado inmobiliario de la Ciudad de Buenos Aires 2017”

**Mentor:** Javier Lezama

**Integrantes Grupo:**

-Navarro Agustina

-Calle Manuel

-Tagle Gabriel

**Agosto 2019**

Introducción

Para la confección del informe se desarrolló una notebook en Jupyter donde se aplicaron los conocimientos de la materia de la diplomatura: “Aprendizaje No Supervisado”.

En la notebook usamos diferentes librerías que nos permitieron combinar herramientas de manejo de grandes volúmenes de datos, análisis estadísticos, división de datos, y algoritmos de predicción. Algunas de las librerías nuevas que hemos utilizado son Kmeans, silhouette\_samples, silhouetes\_scores, StandardScaler, feature\_selection, PCA y resample.

El mentor de este grupo es Javier Lezama quien nos orientó en lo relacionado al contenido del dataset y a la descripción del caso para poder desarrollar este informe.

Definición de los datos del dataset

Para comenzar el análisis, se planteó que el dataset a usar debía ser el original, por lo que definimos del total de variables algunas que consideramos más importantes para poder aplicar los algoritmos del clustering. Se definieron las siguientes:

* Tipo de Propiedad (‘property\_type’)
* Superficie Total en m2 ('surface\_total\_in\_m2')
* Precio en dólares por m2 (‘price\_usd\_per\_m2’)
* Precio aproximado en dolares (‘price\_aprox\_usd’)
* Cantidad de Habitaciones ('rooms')
* Superficie Cubierta en m2 ('surface\_covered\_in\_m2')

Se limpió el dataset de aquellas filas que estaban nulos o sin información alguna de sus características. Se dividió por clases de Tipos de Propiedad para encontrar cuantas clases tenemos definidas y cuantos ejemplos de cada una:

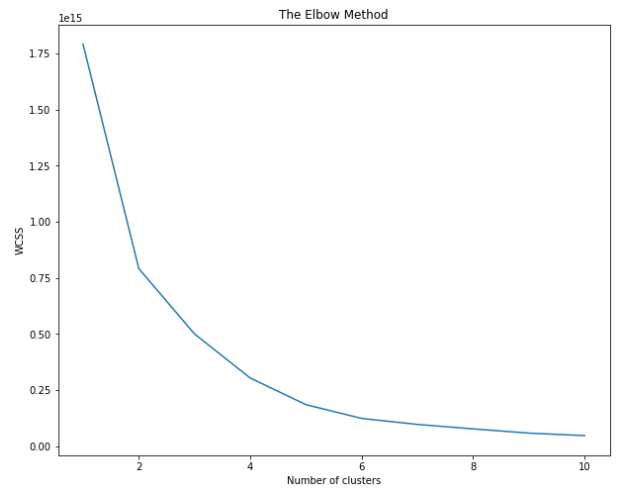
* PH (0) 990
* Apartment (1) 10337
* House (2) 648
* Store (3) 84

Se observó que la cantidad de filas “store” es muy baja por lo que se eliminó para el análisis posterior.

Algoritmo de clustering – K means

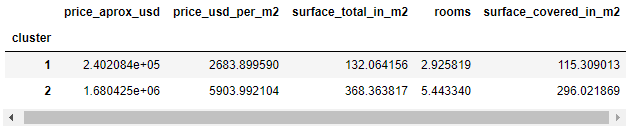
Con el conjunto de features elegidas, se planteó el usar el algoritmo de clustering K-means para observar las clases que se pueden predecir teniendo como ejemplo que son 3 clases (Tipos de Propiedades).

Primero se aplicó el “método del codo” para identificar el punto, que indique cual es la mejor elección de cantidad de clusters:

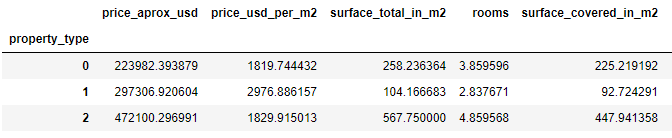


Se observó que un punto de inflexión puede ser elegir 2 clusters. Por lo que se aplicó el análisis de silueta para 2 clusters, obteniendo un silhouete\_score de 0.8496143621819137, con una gran cantidad de ejemplos en una clase “1” (11472) y el resto en otra clase “2” (503).

Se revisó la media de cada una de las variables y se vio que en gran medida que detecto como clase “1” a los que tienen un valor más chico en promedio de precio en dólares, de precio por m2, superficie total en m2 , superficie cubierta y habitaciones.



Esto se revisó en las clases reales, en donde observamos que los valores más bajos de medias para todas las variables son para los apartments (property\_type=1).



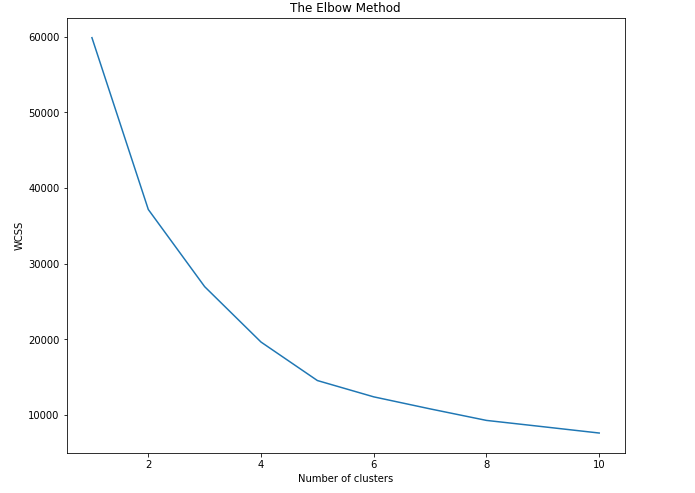
Salvo por “precio\_usd\_per\_m2” el resto de los valores para las houses (2) tienen valores mas elevados de media para compararlos con el cluster 2 obtenido anteriormente.

Por otro lado, se comparó el resultado obtenido de clases y se le acertó al 82.87% del dataset, pero no detectó ningún “PH”. A todos los detectó como de la clase “1” que entendemos que es apartments.

*Estandarización*

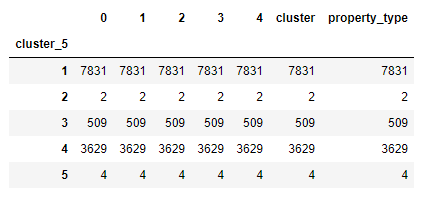
Para un mejor trabajo del algoritmo K-means se sugiere estandarizar las variables por lo que se realizó mediante la librería StandardScaler.

Se volvió a aplicar el “método del codo”, y en este caso se nota menos que cantidad de clusters es la más adecuada para dividir este dataset.



Si aplicamos 2 clusters casi todos los ejemplos son detectados como 1 sola clase (11973 de 11975).

Si aplicamos 5 clusters que puede indicar el metodo del codo, obtenemos 2 clusters bien definidos (7831 ejemplos y 3629 ejemplos), 1 con pocos valores (509) ejemplos y 2 clusters casi sin ejemplos (4 y 2 ejemplos respectivamente).



Se puede entender que la primer clase podrían ser los “apartments”, pero el resto no se puede definir.

Embedings

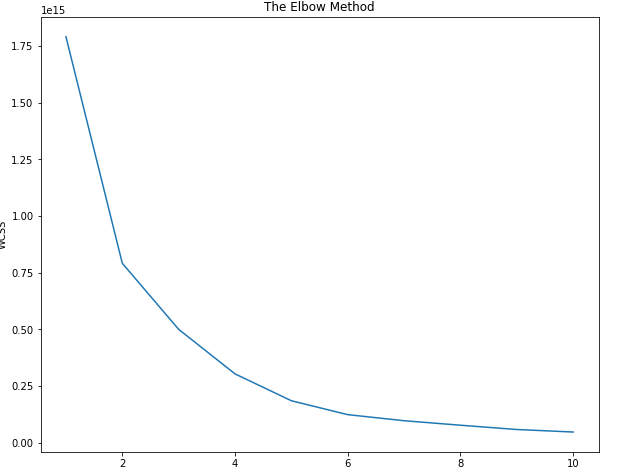
Se aplicaron diferentes embedings para luego aplicar el algoritmo K-means para observar si se pueden obtener mejores resultados.

*Feature Selection*

Se aplicaron 2 métodos:

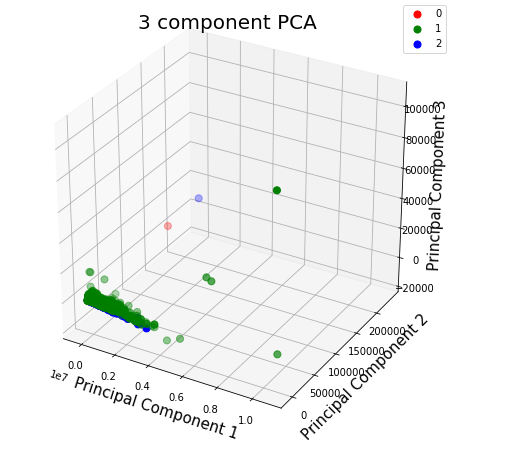
* El de la varianza, para determinar cuáles features tenían una varianza baja, lo que indicaría que influye poco en la definición de los clusters.

Este método detectó que todas las features son importantes para esta definición por lo que no se eliminó ninguna.

* **SelectKBest** para seleccionar las K mejores features. De las 5 features, se optó por quedarnos con las mejores 4. Se eliminó la feature “rooms” y se aplicó el algoritmo k-means con el nuevo dataset, nuevamente detectando 2 clusters (clase 1 = 11472, clase 2 = 503).
* 

*PCA*

Se aplicó este método reduciendo la dimensionalidad a 3 componentes y graficandolas vemos que se encuentran muy “pegadas” y es difícil diferenciarlas.



Lo que se puede notar es que aplicando el algoritmo k-means detectó 2 clases y las dos clases con los mismos ejemplos que el método anterior.

*Balanceo de clases*

Se planteó generar ejemplos en las clases iniciales desbalanceadas, para que todas queden con la misma cantidad de ejemplos para luego aplicar el algoritmo para clusterizar. Para esto se uso la librería resample.

*Oversample*

Consiste en generar ejemplos de más en todas las clases para que queden con la misma cantidad de ejemplos que la clase con más ejemplos.

De este modo previo a clusterizar tenemos misma cantidad de PH, apartment y house, 10337.

*Undersample*

Consiste en eliminar ejemplos de las clases para que queden con la misma cantidad de ejemplos que la clase con menos ejemplos. No es un buen método, ya que al eliminarse al azar se pueden perder particularidades de una clase.

De este modo previo a clusterizar tenemos misma cantidad de PH, apartment y house, 648.

De todos modos, obtenemos 2 clases muy marcadas para ambos casos de balanceo.

Conclusiones

Este trabajo práctico permitió al grupo aplicar conocimientos necesarios para entender el concepto del aprendizaje no supervizado y la dificultad que conlleva el diferenciar un dataset por clases, además del entender por qué un algoritmo dividió en n clases, que es lo que usa el algoritmo para diferenciarlas.

En este trabajo se utilizó el algoritmo de clustering k-means y se aplicaron diferentes técnicas de embendings para trabajar los datos y así poder encontrar clases que se asemejen a lo real, aprovechando que tenemos el resultado de clases a detectar.

Se aplicó el “método del codo” y “silhouette” análisis para identificar el mejor número de clases a elegir y analizar cada resultado, observando que este algoritmo no puede detectar las 3 clases teniendo que conformarnos que la mejor opción es elegir dos clases, una que engloba gran parte del dataset y la otra que se queda con el resto.

Sin embargo, aún se puede seguir indagando aplicando otros algoritmos para determinar estas 3 clases, o detenerse en otra feature, como puede ser “barrios” e intentar aplicar algún algoritmo de clusterización como en este trabajo, pero teniendo en cuenta que en este caso estamos hablando de muchas clases que debería detectar nuestro algoritmo.