Clasificación de acciones mediante k-means clustering

Trabajo final de posgrado QUANT

Universidad del CEMA

Lic. Facundo Joel Allia Fernandez

Diciembre 2022

Índice

Introducción

Este trabajo presenta un enfoque para utilizar el algoritmo ***K-means*** para clasificación de acciones, con el objetivo de contribuir a inversores a diversificar sus portafolios de inversión.

En el **marco teórico** se presenta las bases teóricas necesarias para arribar al concepto del algoritmo de agrupamiento *k-means*. Se profundiza en los pasos de su funcionamiento y se discuten sus ventajas y limitaciones.

En el **apartado práctico** se utiliza el algoritmo para generar clusterizaciones de acciones según su **Retorno medio anualizado** y **Volatilidad media anualizada.** A continuación, se añaden al análisis las variables **PER** (Ratio precio-beneficio) y **Dividend Rate** (rentabilidad por dividendo). Posteriormente, se extiende el análisis a 3 dimensiones mediante el algoritmo ***K-means++,*** clusterizando por Retorno medio anualizado y Volatilidad media anualizada nuevamente e introduciendo el ratio **Price to Book** como nueva dimensión.

Marco teórico

Machine Learning

El aprendizaje de máquina o ML por sus siglas en ingles es *"El estudio de algoritmos de computación que mejoran automáticamente su rendimiento gracias a la experiencia. Se dice que un programa informático aprende sobre un conjunto de tareas, gracias a la experiencia y usando una medida de rendimiento, si su desempeño en estas tareas mejora con la experiencia."* (Mitchell, 1997). En otras palabras, estudia el aprendizaje automático a partir de datos para conseguir hacer predicciones precisas a partir de observaciones con datos previos. La **clasificación** automática de objetos o datos es uno de los objetivos del aprendizaje de máquina. Podemos considerar tres tipos de algoritmos:

- **Clasificación supervisada:** Se dispone de un conjunto de datos llamados datos de entrenamiento asociados a una etiqueta. A través del entrenamiento de un modelo utilizando dichas etiquetas, se determina si una imagen está clasificada correcta o incorrectamente por dicho modelo. Una vez construido el modelo, es posible utilizarlo para clasificar nuevos datos que, en esta fase, ya no necesitan etiqueta para su clasificación, aunque sí la necesitan para evaluar el porcentaje de datos bien clasificados.
- **Clasificación no supervisada:** los datos no tienen etiquetas (o no se desea utilizarlas) y estos se clasifican a partir de su estructura interna (propiedades, características).
- **Clasificación semisupervisada:** algunos datos de entrenamiento tienen etiquetas, pero no todos ellos. Estos se pueden considerar algoritmos supervisados que no necesitan todas las etiquetas de los datos de entrenamiento.

En Machine Learning, el desarrollo de algunos algoritmos ya ha alcanzado cierta madurez, como el del algoritmo KNN, el algoritmo k-Means, etc. Dichos algoritmos se pueden aplicar a las inversiones en acciones y pueden lograr muy buenos resultados. (Zhao Gao 2020).

El algoritmo k-means

El término k-means fue utilizado por primera vez por MacQueen en 1967, aunque la idea se remonta a Steinhaus en 1957. K-means es un algoritmo de clasificación no supervisada (clusterización) que agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre

cada objeto y el centroide de su grupo o cluster. Se suele usar la distancia cuadrática. El algoritmo consta de tres pasos:

1. **Inicialización:** una vez escogido el número de grupos, k, se establecen k centroides en el espacio de los datos, por ejemplo, escogiéndolos aleatoriamente.
2. **Asignación objetos a los centroides:** cada objeto de los datos es asignado a su centroide más cercano.
3. **Actualización centroides:** se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

Se repiten los pasos 2 y 3 hasta que los centroides no se mueven, o se mueven por debajo de una distancia umbral en cada paso.

El algoritmo *k-means* resuelve un **problema de optimización**, siendo la función a optimizar (minimizar) la suma de las distancias cuadráticas de cada objeto al centroide de su cluster. Los objetos se representan con vectores reales de d dimensiones $(x_1,x_2,\ldots,x_n)$ y el algoritmo *k-means* construye k grupos donde se minimiza la suma de distancias de los objetos, dentro de cada grupo $S=\{S_1,S_2,\ldots,S_k\}$, a su centroide. El problema se puede formular de la siguiente forma:

$$\min_s E(\mu_i) = \min_s \sum_{i=1}^{k} \sum_{x_j \varepsilon S_j} \left\| x_j - \mu_i \right\|^2$$

donde S es el conjunto de datos cuyos elementos son los objetos x_j representados por vectores, donde cada uno de sus elementos representa una característica o atributo. Tendremos k grupos o clusters con su correspondiente centroide μ_i.

En cada actualización de los centroides, desde el punto de vista matemático, imponemos la condición necesaria de extremo a la función $E(\mu_i)$ que, para la función cuadrática es:

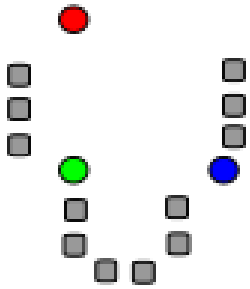
$$\frac{\partial E}{\partial \mu_i} = 0 \Rightarrow \mu_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j$$

y se toma el promedio de los elementos de cada grupo como nuevo centroide. Las principales ventajas del método k-means son que es un método sencillo y rápido. Pero

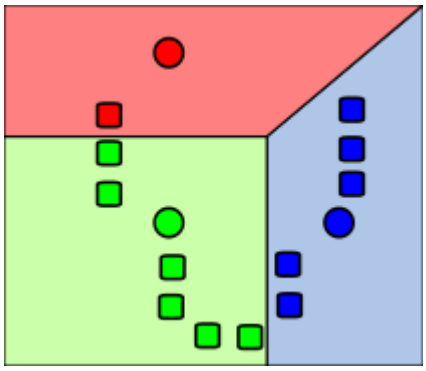
es necesario decidir el valor de *k* y el resultado final depende de la inicialización de los centroides. En principio no converge al mínimo global sino a un mínimo local.

Los métodos de **inicialización de Forgy y Partición Aleatoria** son comúnmente utilizados. El método Forgy elige aleatoriamente k observaciones del conjunto de datos y las utiliza como centroides iniciales. El método de partición aleatoria primero asigna aleatoriamente un clúster para cada observación y después procede a la etapa de actualización, por lo tanto, calcular el clúster inicial para ser el centro de gravedad de los puntos de la agrupación asignados al azar. El método Forgy tiende a dispersar los centroides iniciales, mientras que la partición aleatoria ubica los centroides cerca del centro del conjunto de datos. Según Hamerly y compañía, el método de partición aleatoria general, es preferible para los algoritmos tales como los k-medias armonizadas y fuzzy k-medias. Para *expectation maximization* y el algoritmo estándar el método de Forgy es preferible.

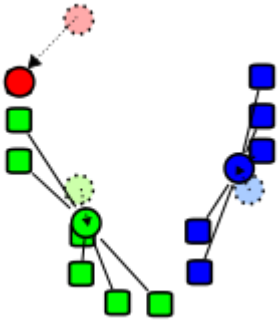
Demostración del algoritmo estándar:



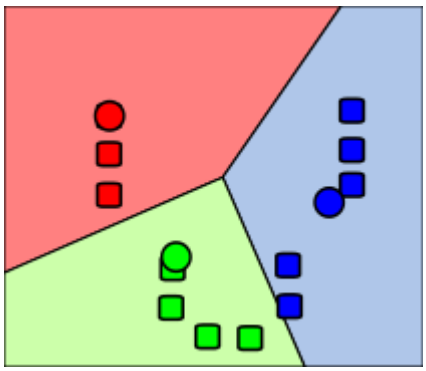
1) *k* centroides iniciales (en este caso *k*=3) son generados aleatoriamente dentro de un conjunto de datos (mostrados en color).



2) *k* grupos son generados asociándole el punto con la media más cercana. La partición aquí representa el diagrama de Voronoi generado por los centroides.



3) EL centroide de cada uno de los *k* grupos se recalcula.



4) Pasos 2 y 3 se repiten hasta que se logre la convergencia.

Clusterización mediante retorno y volatilidad

Para llevar a cabo la clusterización se trabaja con la totalidad de acciones que componen el índice S&P 500. El S&P 500 consta de 500 empresas que representan todos los sectores de la economía. El índice cubre solo empresas de gran capitalización que cotizan en el mercado estadounidense, ya sea en la Bolsa de Valores de Nueva York o en el Nasdaq. Debido a que el S&P 500 representa a las empresas más grandes que cotizan en bolsa en los EE. UU., se considera uno de los índices bursátiles más representativos, por lo tanto, constituye un buen set de datos para el análisis de clusterización por algoritmo k-means.

Una vez obtenidos los precios de cierre ajustados para el periodo bajo estudio (02/01/2020 – 02/12/2022) de las 500 compañías que componen el índice es posible obtener el retorno anual promedio de cada una, así como su volatilidad media anualizada. Para obtener los datos de los retornos, los insertamos en un *dataframe* y luego los anualizamos (se estima que un año tiene 252 días de mercado).

$$\bar{X} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$Retorno\ medio\ anualizado = \bar{X} * 252$$

Posteriormente se calcula la volatilidad media anualizada a través de la formula:

$$\sigma_{\bar{X}} = \sqrt{Var(\sum_{i=1}^{n} x_i)}$$

$$Volatilidad\ media\ anualizada = \sigma_{\bar{X}} * \sqrt{252}$$

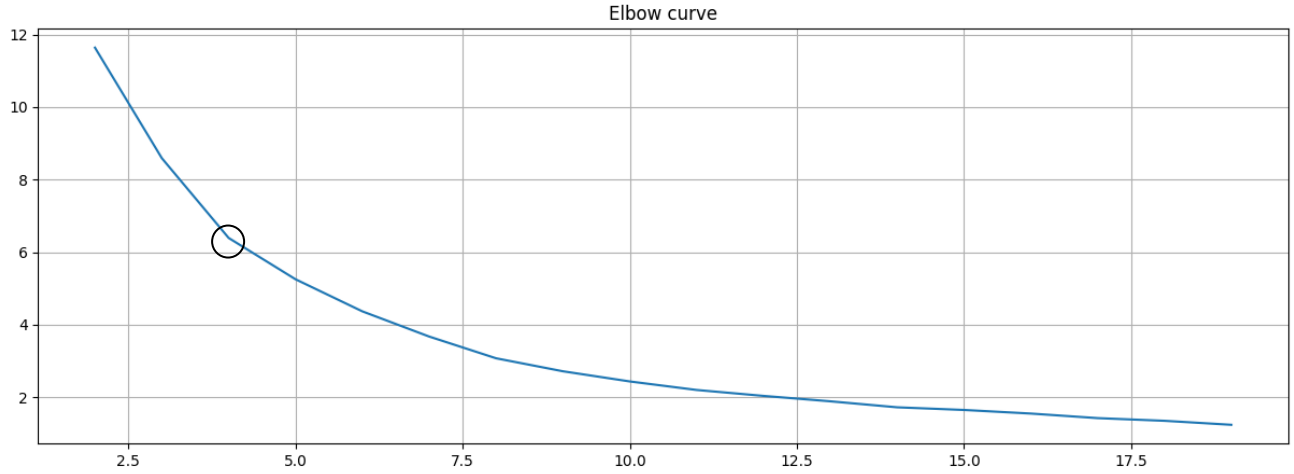
Uno de los problemas que surgen a la hora de aplicar el método de Clustering K-means es la elección del número de Clusters. No existe un criterio objetivo ni ampliamente válido para la elección de un número óptimo de Clusters; pero se debe tener en cuenta, que una mala elección de los mismos puede dar lugar a realizar agrupaciones de datos muy heterogéneos (pocos clusters); o datos, que siendo muy similares unos a otros los agrupemos en Clusters diferentes (muchos Clusters).

Para determinar el número óptimo de clusters k para el conjunto de datos, se utiliza el Método del codo (Elbow Method). Este método utiliza los valores de la inercia obtenidos tras aplicar el K-means a diferente número de Clusters (desde 1 a N

Clusters), siendo la inercia la suma de las distancias al cuadrado de cada objeto del clúster a su centroide:

Una vez obtenidos los valores de la inercia tras aplicar el K-means de 1 a N Clusters, se representan en una gráfica lineal la inercia respecto del número de Clusters. En esta gráfica se aprecia un cambio brusco en la evolución de la inercia, teniendo la línea representada una forma similar a la de un brazo y su codo. El punto en el que se observa ese cambio brusco en la inercia nos dirá el número óptimo de Clusters a seleccionar para ese data set, dicho de otra manera: el punto que representaría al codo del brazo será el número óptimo de Clusters para ese data set.

Para el modelo bajo estudio se ajustan diferentes modelos del algoritmo K-means mientras se varía el parámetro k en el rango de 2 a 20. Para cada modelo se calcula el error de suma cuadrática (SSE) utilizando el método de inercia del modelo ajustado. Se selecciona el modelo con el menor valor de SSE. (La inercia indica qué tan lejos están los puntos dentro de un grupo. Cuanto menor sea el valor de la inercia, mejor desempeño tendría el modelo).



Es posible observar que una vez que el número de grupos llega a 4 (en el eje inferior), la reducción en el SSE comienza a disminuir por cada aumento en el número de grupos. Esto llevaría a creer **que el número óptimo de grupos para el modelo se encuentra alrededor de 4.**

Una vez definido el numero optimo de clusters se procede a crear los mismos. Para lo cual, en primera instancia se definen los centroides utilizando la librería *sklearn*. Para de la creación de 4 grupos de acciones, el algoritmo K-means asigna iterativamente puntos de datos a los grupos en función de su similitud de características, o "*features*", en este caso **Retorno medio anualizado** y **Volatilidad media anualizada**.

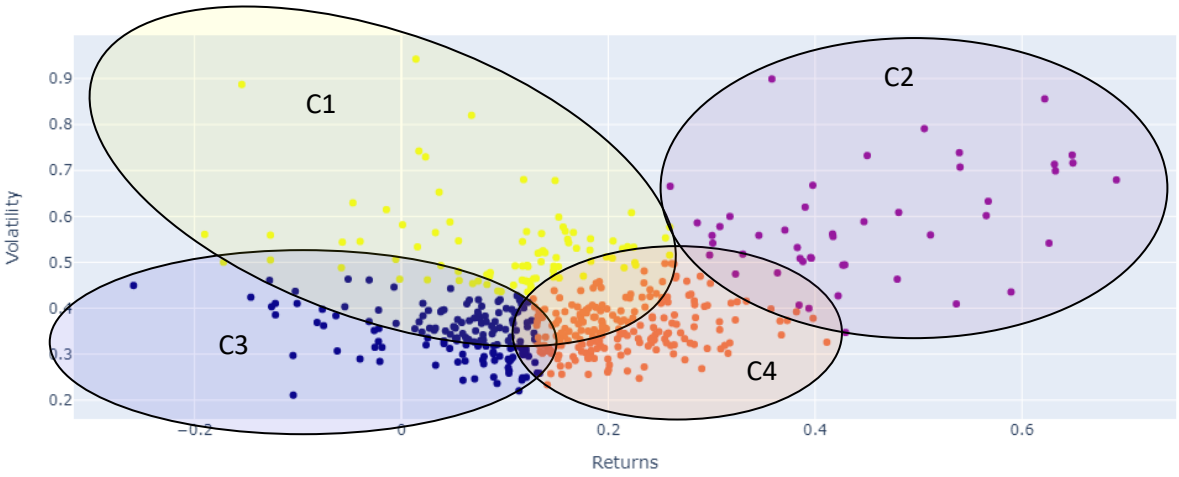
El algoritmo inicialmente asigna los puntos de datos aleatoriamente a los grupos y luego calcula el centroide de cada grupo, que es la media de todos los puntos de datos dentro del grupo. Luego, compara los puntos de datos con el centroide y los reasigna a grupos según corresponda. Este proceso se repite hasta que el centroide de cada grupo permanece relativamente estable, momento en el cual el algoritmo se detiene y a cada grupo se le asigna una etiqueta. El resultado final es un conjunto de 4 grupos, cada uno de los cuales contiene acciones que tienen rendimientos y volatilidades similares.

Al crear los clusters, se detectan cuatro **valores atípicos** o *outliers* en un gráfico de dispersión. Los *outliers* son puntos de datos que son significativamente diferentes del resto de los puntos de datos en el conjunto de datos. A menudo, pueden conducir a resultados inexactos cuando se usa un algoritmo, ya que no encajan en el mismo patrón que los otros puntos de datos. Por lo tanto, es importante segregar y eliminar los valores atípicos para mejorar la precisión del modelo.

La eliminación de *outliers* puede ayudar al algoritmo a centrarse en los puntos de datos más representativos y reducir el efecto de los valores atípicos en los resultados. Esto puede ayudar a aumentar la precisión del modelo y garantizar que los puntos de datos estén agrupados correctamente. Los tickers eliminados son:

Ticker	Retorno medio anualizado	Volatilidad media anualizada
MRNA	1.1293	0.8544
ENPH	1.1729	0.8290
TSLA	0.9144	0.7185
CEG	1.0181	0.4901

Una vez eliminados los valores atípicos, repetimos los pasos realizados para el agrupamiento utilizando el algoritmo K-means para obtener agrupaciones más precisas. El resultado de aplicar el algoritmo k-means son 4 clusters constituidos por acciones con similar retorno medio anualizado y volatilidad media anualizada:



El gráfico muestra 4 clusters que fueron generados mediante un algoritmo K-means con 2 variables: retorno medio anualizado y volatilidad media anualizada. Estas variables se usan para medir el riesgo y el rendimiento de una acción. Los 4 clusters representan 4 grupos de acciones con diferentes niveles de riesgo y retorno en el periodo bajo estudio.

clúster	Retorno medio anualizado	Volatilidad media anualizada
C1	Bajo o negativo	Alta
C2	Alto	Alta
C3	Bajo o negativo	Baja
C4	Medio	Baja

La clusterización es útil para identificar grupos de similares entre las acciones, permitiendo así diferenciar entre las acciones con diferentes niveles de riesgo y retorno. Esto es útil para los inversores que buscan diversificar sus portafolios de inversión, ya que les permite identificar grupos de acciones con diferentes niveles de riesgo y retorno.

Los inversores podrían utilizar los 4 clusters para seleccionar una combinación de acciones con diferentes niveles de riesgo y retorno en función de sus objetivos de inversión. Esto les ayudará a diversificar su portafolio y a reducir el riesgo de su inversión, ya que estarán invirtiendo en una variedad de activos con diferentes niveles de riesgo.

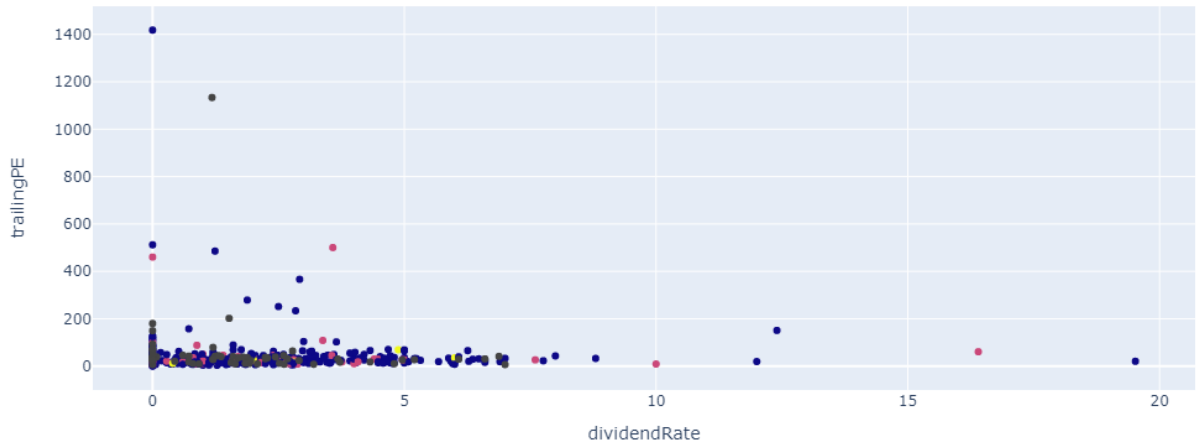
Clusterización mediante P/E y tasa de dividendos

Según Zhao Gao: *En machine learning, el desarrollo de algunos algoritmos ya es bastante maduro, […] como algoritmo k-Means. Dicho algoritmo puede ser aplicado a inversiones en renta variable y lograr muy buenos resultados. Por ejemplo, segregando dos tipos de empresas en el mercado. Por un lado, empresas maduras o "value stocks" que, por lo general, tienen relaciones P/E bajas y altas tasas de dividendos. La segunda categoría, compañías "growth" son empresas con amplias perspectivas de desarrollo, pero también incertidumbres en el futuro, generalmente tienen relaciones P/E altas y tasas de dividendos bajas. Si puede distinguir con precisión entre acciones de primera clase y acciones de alto crecimiento en el mercado, puede proporcionar una buena referencia para los inversores.* (Zhao Gao 2020).

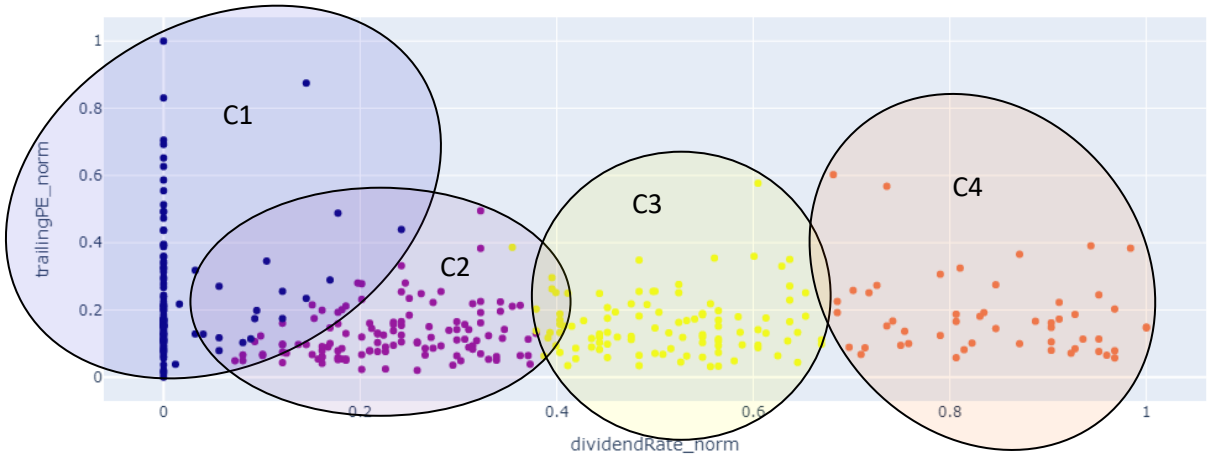
Siguiendo esta línea conceptual es posible aplicar una clusterización similar a la realizada previamente intercambiando las variables Retorno medio anualizado y Volatilidad media anualizada por **PER** (Ratio precio-beneficio) y **Dividend Rate** (rentabilidad por dividendo). De esta manera podríamos diferenciar entre compañías *"value"* y compañías *"growth"*.

Para tal fin, se importan los datos *trailing price-to-earnings (P/E)* y *dividend rate*. La relación precio-beneficio móvil o *trailing price-to-earnings (P/E)*, es un múltiplo de valoración relativa que se basa en los últimos 12 meses de ganancias reales. Se calcula tomando el precio actual de las acciones y dividiéndolo por las ganancias por acción *(BPA)* de los últimos 12 meses. Mientras que la tasa de dividendos o *dividend rate*, es la cantidad de efectivo que una empresa devuelve a sus accionistas anualmente como porcentaje del valor de mercado de la empresa.

Al realizar una primera aproximación clusterizando por *trailing price-to-earnings (P/E)* y *dividend rate* se evidencia la presencia de outliers y excesiva dispersión entre las observaciones, por lo que se procede a filtrar las acciones y normalizar los datos para eliminar estas distorsiones.



Una vez realizadas las modificaciones pertinentes, es posible obtener 4 clusters generados por el algoritmo K-means según el *trailing price-to-earnings (P/E)* y *dividend rate* de cada acción.



Es posible comprobar gráficamente que el algoritmo asigno mayor ponderación a la variable *dividend rate* a la hora de crear los clusters. Se distinguen de esta manera 4 conjuntos de acciones**:** **C1** con Nula o muy baja, **C2** con baja, **C3** con una media-alta y **C4** con alta *dividend rate.*

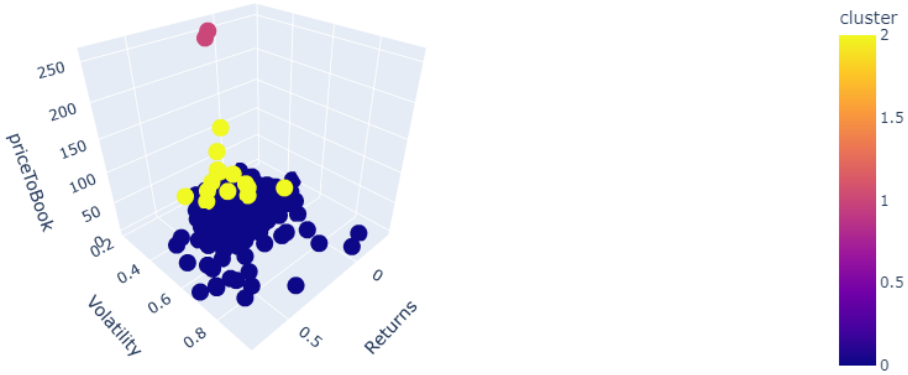
Agrupamiento Tridimensional Con K-means++

Podemos ampliar el análisis de las acciones del S&P500 aplicando el agrupamiento k-means++. Este algoritmo asegura una inicialización más inteligente de los centroides y mejora la calidad del agrupamiento. Aparte de la inicialización, el resto del algoritmo es el mismo que el algoritmo estándar de K-means. Es decir, K-means++ es el algoritmo estándar de K-means junto con una inicialización más inteligente de los centroides.

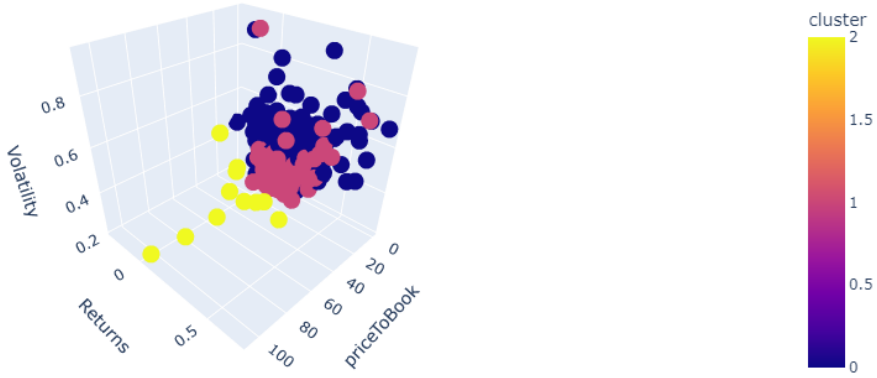
Es posible tomar en cuenta 3 variables para la clusterización. En la clusterización con K-means, los puntos se agrupan en clusters basados en la distancia entre los puntos. Esto significa que para extender una clusterización de dos dimensiones a tres, se debe agregar una tercera dimensión a los datos y calcular la distancia entre los puntos también en esa dimensión.

Los clústeres se definen ahora como conjuntos de datos que tienen distancias similares en todas las tres dimensiones, en lugar de sólo dos. Esto permite una mejor separación de los clústeres y un mejor ajuste a la distribución de los datos.

Al igual que en la primera aplicación, se seleccionan los datos de Rentabilidad media anualizada y Volatilidad media anualizada, pero ahora además se añade la variable **Price to Book** para el análisis en 3 dimensiones:



Nuevamente, notamos la presencia de outliers o valores atípicos. Se procede a eliminarlos individualmente y se repiten los pasos para volver a agrupar.



Finalmente, se obtiene por medio de la clusterización por algoritmo K-means++ 3 conjuntos de acciones agrupadas por las 3 variables bajo estudio (Rentabilidad media anualizada, Volatilidad media anualizada y Price to Book). Es llamativa la presencia de acciones con altos índices Price to Book. Esta información puede ser de utilidad a la hora de conformar un portafolio de inversión.

Bibliografía

- Mitchell, T. (1997). Machine Learning (1.ª ed.). McGraw-Hill Education.
- K-means. (s. f.). Recuperado 26 de octubre de 2022, de https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html
- Zhao Gao (2020). The application of artificial intelligence in stock investment. J. Phys.: Conf. Ser. 1453 012069
- Jamieson (2018). K-Means Algorithm Python Example https://www.pythonforfinance.net/2018/02/08/stock-clusters-using-k-means-algorithm-in-python/
- Shravan SK (2020). K-means for 3 variables https://medium.com/@sk.shravan00/k-means-for-3-variables-260d20849730
- Shivangi Singh (2021). K Means Clustering on High Dimensional Data https://medium.com/swlh/k-means-clustering-on-high-dimensional-data-d2151e1a4240