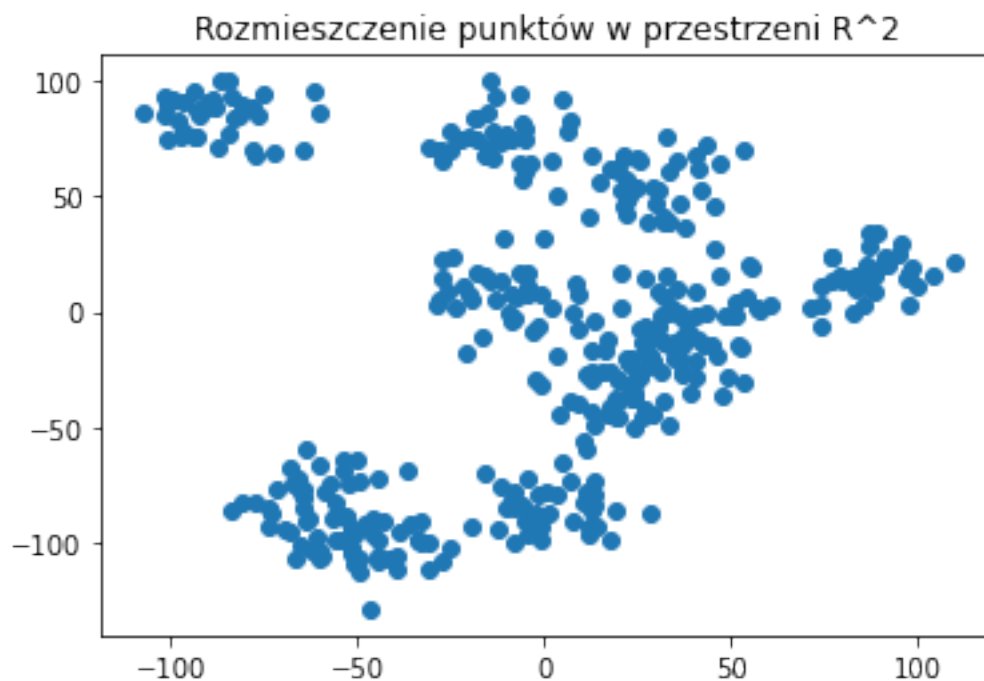


May 18, 2021

```
[1]: import pandas as pd
import matplotlib.pyplot as plt

clusters_df = pd.read_csv('clustering.csv', names = ["X", "Y"], header = None)
plt.scatter(clusters_df.X, clusters_df.Y)
plt.title("Rozmieszczenie punktów w przestrzeni R^2")
plt.show()
```



1 Kmeans z metodą łokcia

```
[2]: # wyznaczymy optymalną liczbę klastrów

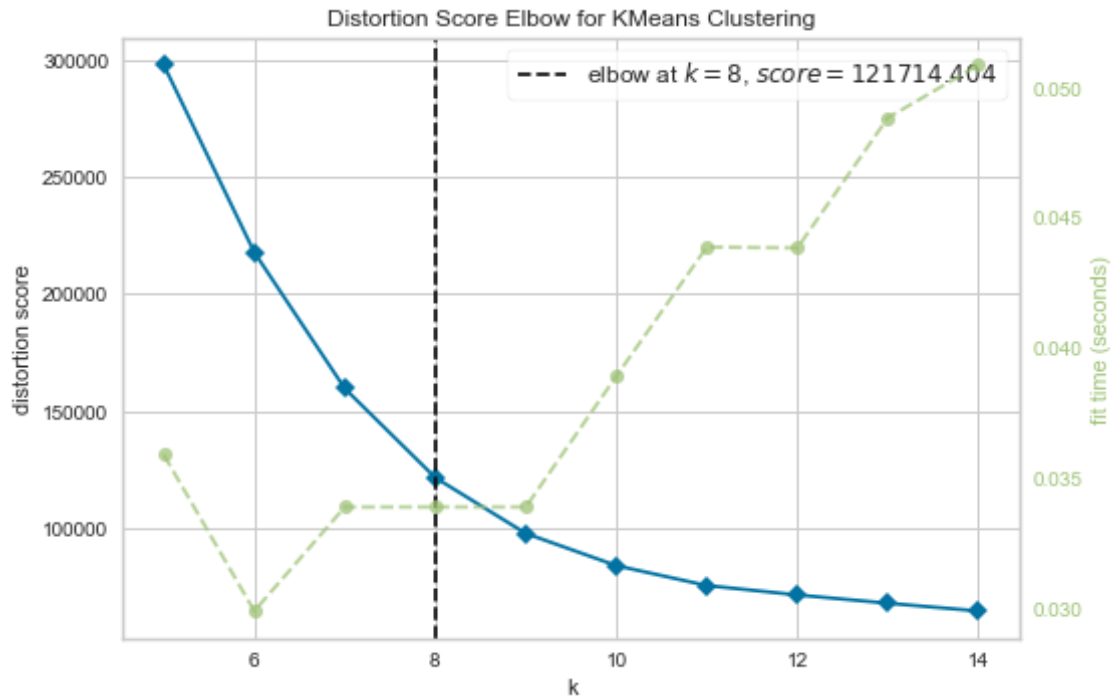
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
```

```

model = KMeans(random_state = 152)
visualizer = KElbowVisualizer(model, k = (5, 15))

visualizer.fit(clusters_df)
visualizer.show()

```



```

[2]: <AxesSubplot:title={'center':'Distortion Score Elbow for KMeans Clustering'},
      xlabel='k', ylabel='distortion score'>

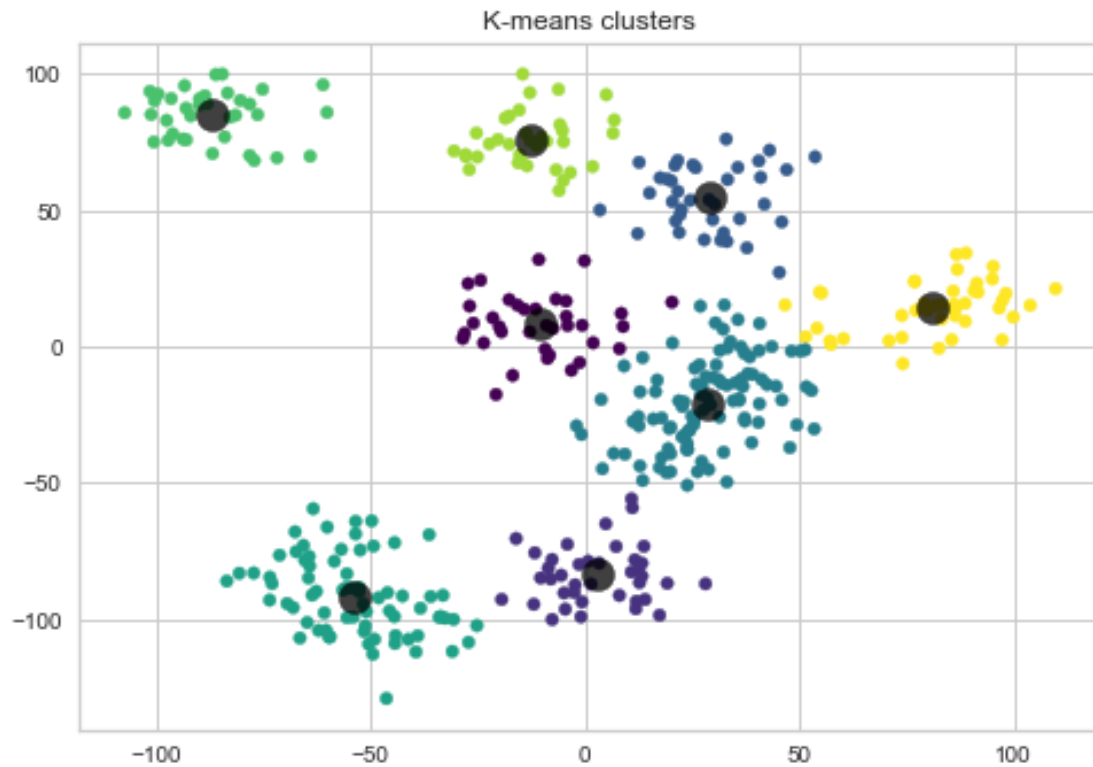
```

```

[3]: def plot_kmeans_clusters(X, n_clusters):
      kmeans = KMeans(n_clusters=n_clusters, random_state=0)
      kmeans.fit(X)
      y_kmeans = kmeans.predict(X)
      plt.scatter(X.iloc[:, 0], X.iloc[:, 1], c=y_kmeans, s=30, cmap='viridis')

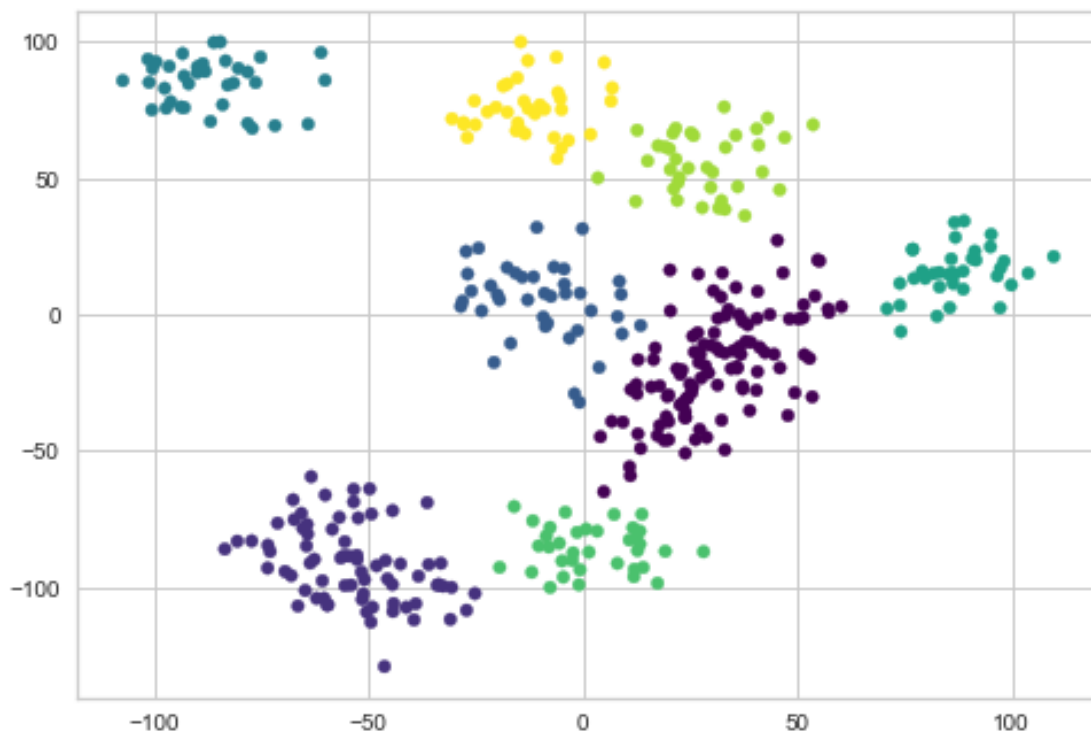
      centers = kmeans.cluster_centers_
      plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.75)
      plt.title('K-means clusters')
      plt.show()
      plot_kmeans_clusters(clusters_df, n_clusters=8)

```



```
[4]: from sklearn.cluster import AgglomerativeClustering

model = AgglomerativeClustering(n_clusters=8)
y = model.fit_predict(clusters_df)
plt.scatter(clusters_df.iloc[:, 0], clusters_df.iloc[:, 1], c=y, s=30, ↵
            cmap='viridis')
plt.show()
```

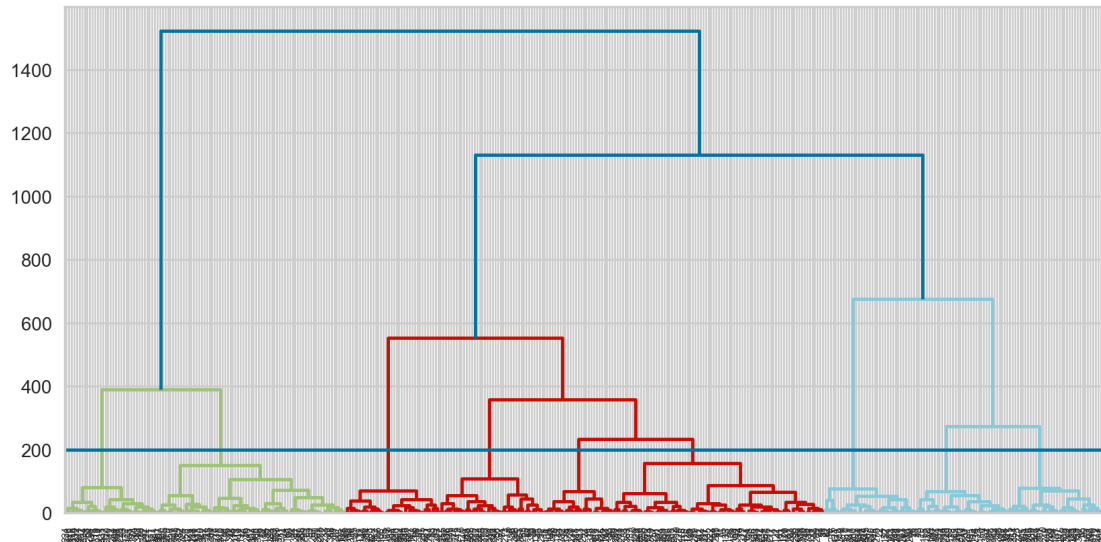


Funkcja wyznaczyła, że metoda łokciowa wyznacza 8 klastrow. Wizualnie widzimy, że istotnie jest to dobra ilość klastrow.

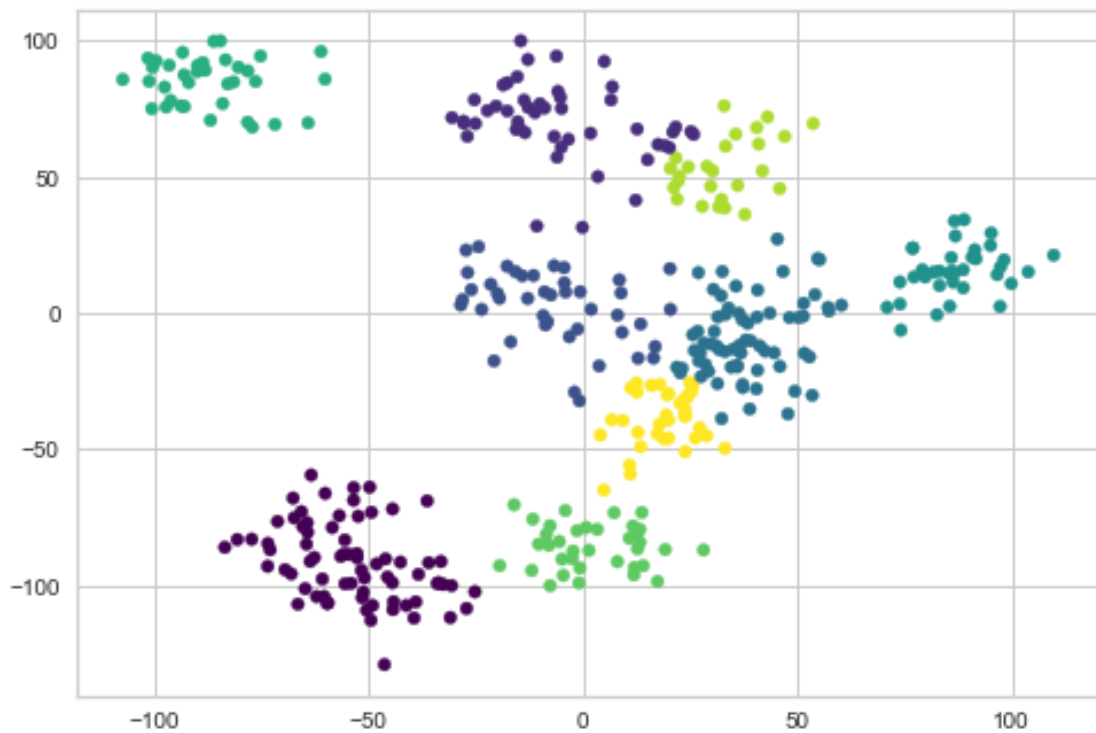
2 Metoda aglomeracyjna

```
[5]: from scipy.cluster import hierarchy

Z = hierarchy.linkage(clusters_df, method='ward')
plt.figure(figsize=(10, 5), dpi= 200, facecolor='w', edgecolor='k')
hierarchy.dendrogram(Z)
plt.axhline(y = 200)
plt.show()
```



```
[6]: model = AgglomerativeClustering(n_clusters=9, linkage='complete')
y = model.fit_predict(clusters_df)
plt.scatter(clusters_df.iloc[:, 0], clusters_df.iloc[:, 1], c=y, s=30,
            cmap='viridis')
plt.show()
```



Metoda aglomeracyjna podzieliła nasz zbiór na 9 klastrow. Widzimy jednak, że podział uzyskany w przypadku metody Kmeans (8 klastrow) był lepszy.

[]: