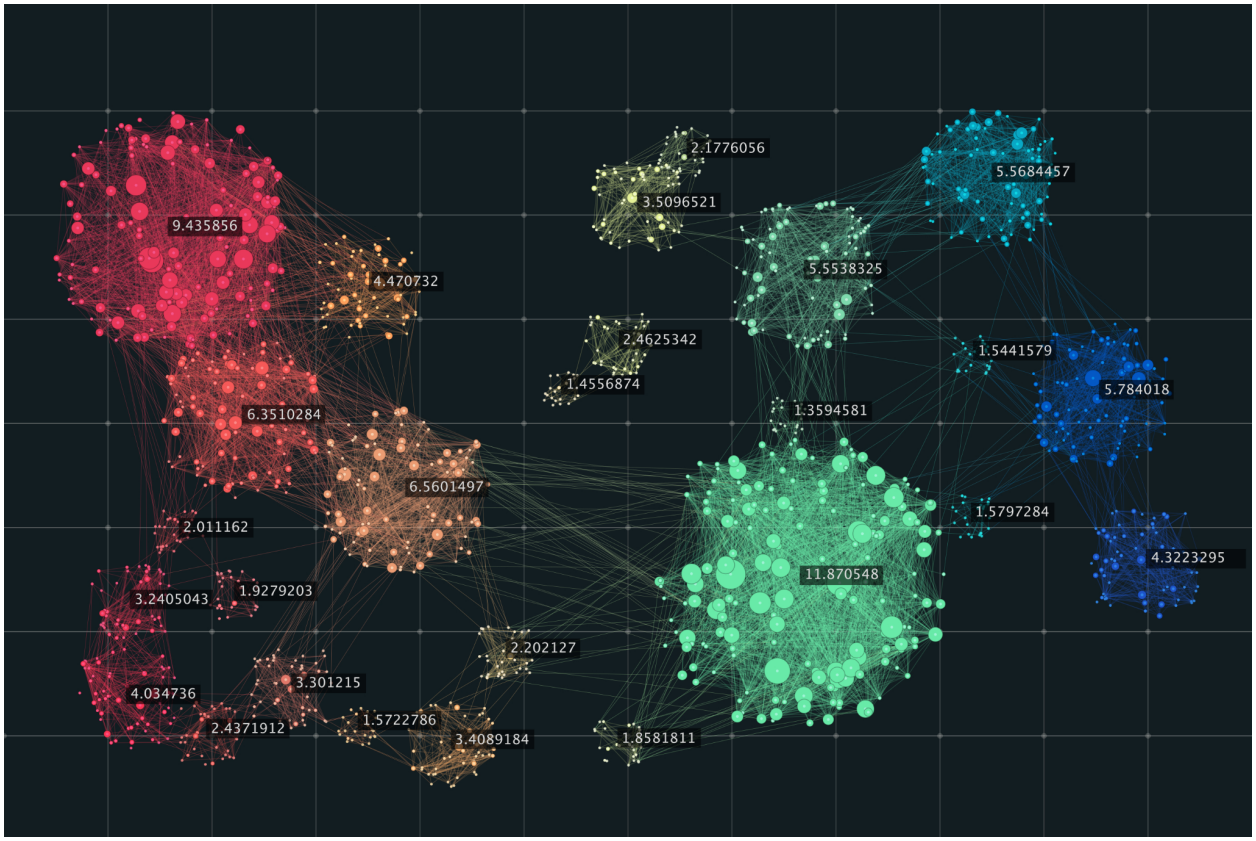
Mall Customer Segmentation

Ομαδοποίηση δεδομένων χρησιμοποιώντας τον αλγόριθμο KMeans



Αντόν Πάπα, 1054337, papa@ceid.upatras.gr

Ιωάννης Σπίγγος, 1064045, st1064045@ceid.upatras.gr

Πρόβλημα

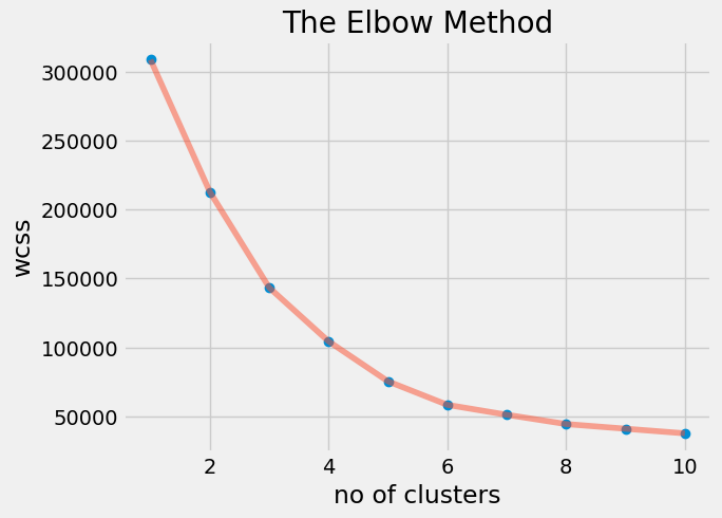
Ο σκοπός αυτής της άσκησης είναι να αποκαλύψει τα υποκείμενα μοτίβα στην πελατειακή βάση και σε ομάδες πελατών αντίστοιχα, το οποίο συχνά είναι γνωστό και ως τμηματοποίηση της αγοράς. Με αυτόν τον τρόπο, η ομάδα μάρκετινγκ μπορεί να έχει μια πιο στοχευμένη προσέγγιση για να προσελκύσει τους καταναλωτές και το εμπορικό κέντρο μπορεί να λάβει πιο ενημερωμένες στρατηγικές αποφάσεις για να αυξήσει τα κέρδη. Αυτό πρέπει να επιτευχθεί χρησιμοποιώντας μη εποπτευόμενη μηχανική μάθηση και αντλώντας δεδομένα από ένα dataset με πληροφορίες πελατών. Το dataset που μας δίνεται είναι μια βάση δεδομένων πελατών ενός εμπορικού κέντρου και περιέχει 200 διαφορετικούς πελάτες με βασικές πληροφορίες όπως ηλικία, φύλο, ετήσιο εισόδημα και βαθμολογία δαπανών.

Μέθοδοι

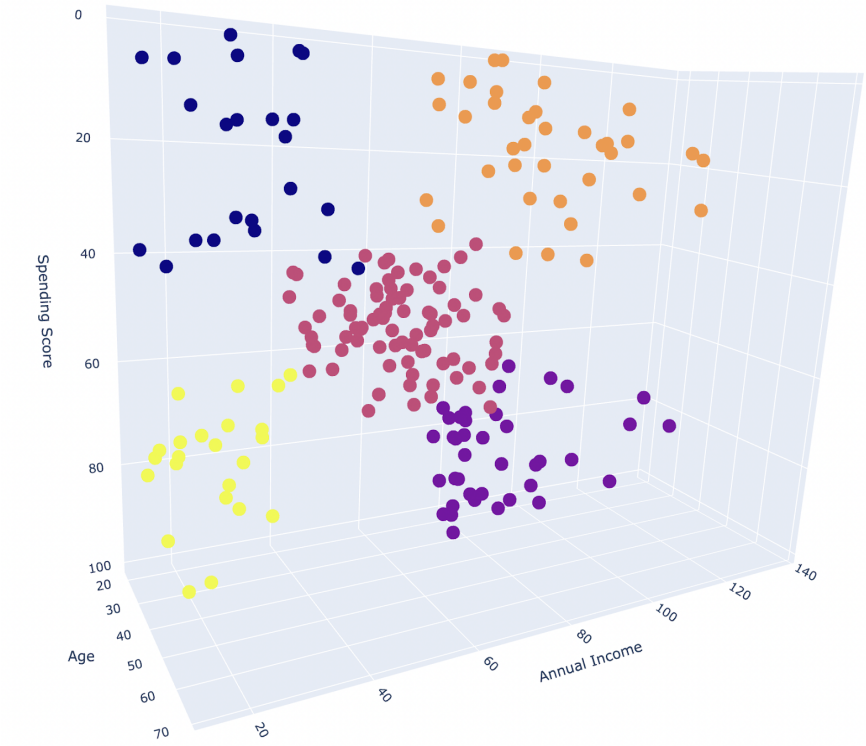
Αρχικά με την βοήθεια της βιβλιοθήκης pandas, διαβάζεται το dataset, δημιουργείται ένα dataframe και μετά από επεξεργασία καταλήγει να περιέχει μόνο τις στήλες “Age”, “Spending Score”, και “Annual Income”, οι οποίες είναι αυτές που χρησιμοποιούνται στην υπόλοιπη άσκηση. Στη συνέχεια με την βοήθεια της βιβλιοθήκης sklearn.cluster χρησιμοποιείται ο αλγόριθμος k-Means, ο οποίος είναι ένας αλγόριθμος μη εποπτευόμενης μηχανικής μάθησης και γίνεται ομαδοποίηση για τιμές του k από 1 έως 10, ώστε να βρεθεί η βέλτιστη τιμή του k. Το k συμβολίζει τον αριθμό των ομάδων που δημιουργούνται και για να βρεθεί ο βέλτιστος αριθμός ομάδων k, χρησιμοποιείται η μέθοδος elbow. Στην μέθοδο αυτή υπολογίζεται το wcss το οποίο είναι το άθροισμα της τετραγωνικής απόστασης μεταξύ κάθε σημείου και του κέντρου της κάθε ομάδας. Το wcss ξεκινά έχοντας την μέγιστη τιμή όταν k=1 και στη συνέχεια, όσο το k αυξάνεται, το wcss αρχίζει να μειώνεται. Αν σχεδιάσουμε το wcss συναρτήσει του k, το διάγραμμα κάθε φορά μοιάζει με αγκώνα, καθώς υπάρχει κάποιο συγκεκριμένο k στο οποίο το wcss μειώνεται απότομα και αρχίζει στη συνέχεια να κινείται σχεδόν παράλληλα με τον άξονα x. Η τιμή k που αντιστοιχεί σε αυτό το σημείο είναι η βέλτιστη τιμή k δηλαδή ο βέλτιστος αριθμός ομάδων που πρέπει να τμηματοποιηθούν τα δεδομένα κατά την εκτέλεση του αλγορίθμου k-Means. Η μέθοδος elbow πραγματοποιείται με την βοήθεια της βιβλιοθήκης matplotlib.pyplot και στη συνέχεια εκτελείται πάλι ο αλγόριθμος k-Means με σκοπό την ομαδοποίηση των δεδομένων, αλλά αυτή την φορά χρησιμοποιώντας το βέλτιστο k. Τέλος, με την βοήθεια της βιβλιοθήκης plotly.graph_objects δημιουργείται ένα τρισδιάστατο διάγραμμα το οποίο αναπαριστά την ομαδοποίηση, αλλά και τα κέντρα που υπολόγισε ο αλγόριθμος k-Means, χρησιμοποιώντας διαφορετικά χρώματα για την κάθε ομάδα.

Αποτελέσματα

Από τις παραπάνω μεθόδους προέκυψε το εξής διάγραμμα elbow method που αναπαριστά το wcss συναρτήσει του k:



Και το εξής τρισδιάστατο διάγραμμα από την αναπαράσταση των ομάδων, αλλά και των κέντρων τους που δημιουργήθηκαν από τον αλγόριθμο k-Means:



Κώδικας

Ο κώδικας υπάρχει και στο GitHub στον εξής σύνδεσμο:
https://github.com/hypercurious/mall-customer-segmentation

```
#dataset to dataframe library
import pandas as pd
#plotting libraries
import matplotlib.pyplot as plt
import plotly.graph_objects as go


'''Reading dataset and selecting age, annual income and spending score columns'''
dataset = pd.read_csv(r'Dataset 3.csv')
X = dataset.iloc[:, 2:5].values

'''Using KMeans clustering to find the optimal number of clusters'''
#import KMeans algorithm
from sklearn.cluster import KMeans
#create within-cluster sum of squares (inertia) list
wcss = []
for i in range(1,11):
   kmeans = KMeans(n_clusters= i, init='k-means++', random_state=0)
   #compute KMeans clustering
   kmeans.fit(X)
   #add KMeans inertia to wcss list
   wcss.append(kmeans.inertia_)

'''Elbow method visualization'''
#change default style to fivethirtyeight style
plt.style.use('fivethirtyeight')
#plot number of clusters (x axis) and wcss (y axis)
plt.plot(range(1,11), wcss, 'o')
plt.plot(range(1,11), wcss, '-', alpha=0.5)
#add plot info
plt.title('The Elbow Method')
plt.xlabel('no of clusters')
plt.ylabel('wcss')
#save and show plot
plt.savefig('figs/elbow.png', bbox_inches='tight')
plt.show()
```

```
'''Model building using KMeans clustering with 5 clusters (k=5 is the optimal number
of clusters according to elbow method)'''
kmeansmodel = KMeans(n_clusters= 5, init='k-means++', random_state=0)
kmeansmodel.fit_predict(X)
labels = kmeansmodel.labels_
centroids = kmeansmodel.cluster_centers_

'''Clusters 3d visualization using graph objects'''
#find the data trace
trace = go.Scatter3d(name='data', text=dataset.iloc[:, 0].values, hoverinfo='text',
x=dataset['Age'], y=dataset['Spending Score (1-100)'], z=dataset['Annual Income
(k$)'], mode='markers', marker=dict(color=labels))
#find the centroids trace
ctrace = go.Scatter3d(name='centroids', x=centroids[:, 0], y=centroids[:, 1],
z=centroids[:, 2], opacity=0.5, mode='markers', marker=dict(size=15, color='black'))
#find the layout given the title and scene options
layout = go.Layout(title='Clusters', scene=dict(xaxis = dict(title  = 'Age'), yaxis =
dict(title  = 'Spending Score'), zaxis = dict(title  = 'Annual Income')))
#create 3d figure using trace and layout
fig = go.Figure(data=[trace, ctrace],layout=layout)
#save and show 3d figure
fig.write_html('figs/3dmodel.html')
fig.show()
```