Project FrailSafe ΥΔΑ

Πετράκης
Κωνσταντίνος

Preprocessing

- Unbalanced Dataset
  - Non-Frail (0): 213
  - Pre-Frail (1): 227
  - Frail (2): 100

```python
# dimiourgw antigrafa twn deigmatvn tis deuteris klasis gia na lavw 200 deigmata
x_copies = list()
y_copies = list()
for i in range(len(y)):
#     gia tin klasi 2
    if y[i]==2:
        x_copies.append(x[i])
        y_copies.append(y[i])
# prosthiki twn diplotipwn digmatwn sto dataset
x = np.append(x,np.array(x_copies),axis=0)
y = np.append(y,np.array(y_copies))
```

```python
# antikathistw ta nan me to meso oro kathe stilis
df_clinical.fillna(df_clinical.mean(),inplace=True)
```

```python
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
scaler = StandardScaler().fit(x_train)
x_train_scaled = scaler.transform(x_train)
x_test_scaled = scaler.transform(x_test)
```
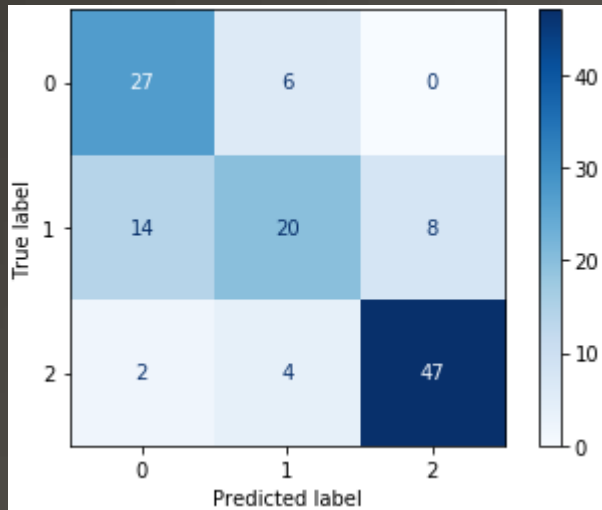
SVM results
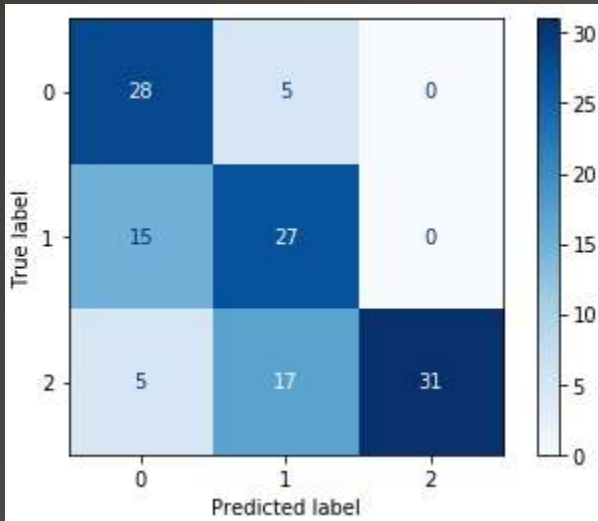
- Accuracy: 67.1875%
- Balanced Accuracy: 69.2082%
- Precision: 71.1451%
- Recall: 69.2082%

```
# SVM CLASSIFIER
clf = svm.SVC(kernel='poly',gamma='auto',decision_function_shape='ovr',probability=True,degree=3,class_weight=None)
clf.fit(x_train_scaled,y_train)
```
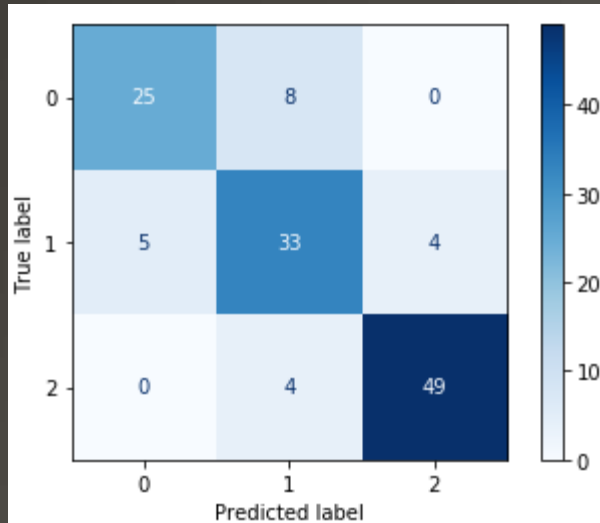
K-NN results

- Accuracy: 73.4375%
- Balanced Accuracy: 72.7055%
- Precision: 71.6373%
- Recall: 72.7055%

```
# K-NN CLASIFIER
clf = neighbors.KNeighborsClassifier(10,weights='distance')
clf.fit(x_train_scaled,y_train)
```

Random Forests results

- Accuracy: 83.59375%
- Balanced Accuracy: 82.2606%
- Precision: 83.0398%
- Recall: 82.2606%

```
# RANDOM FOREST CLASSIFIER
clf = RandomForestClassifier(n_estimators=12, max_depth=None,bootstrap=False) # random_state=1
clf.fit(x_train_scaled,y_train)
```

Συγκεντρωτικά Αποτελέσματα

Μετρική	SVM	K-NN	RandomForests
Accuracy	67.18%	73.43%	**83.59%**
Balanced_Accuracy	69.20%	72.70%	**82.26%**
Precision	71.14%	71.63%	**83.03%**
Recall	69.20%	72.70%	**82.26%**

Preprocessing beacons

- 46782 δείγματα μετά και την αφαίρεση των λανθασμένων part_id

- Μετακίνηση: 5sec.
  - Αλλαγή δωματίου σε <5sec θεωρείται σφάλμα των sensors.

- End_time - begin_time - 5 ο χρόνος που πέρασε στο δωμάτιο με timestamp begin_time.

Clustering

```python
x = np.array(df)
x = Normalizer().fit_transform(x)
```

```python
kmeans = KMeans(n_clusters=3).fit(x)
kmeans_labels = kmeans.labels_
ag = AgglomerativeClustering(n_clusters=3,linkage='single').fit(x)
ag_labels = ag.labels_
spectral = SpectralClustering(n_clusters=3,affinity='precomputed').fit(laplacian_kernel(x))
laplacian_labels = spectral.labels_
spectral = SpectralClustering(n_clusters=3,affinity='nearest_neighbors').fit(x)
spectral_nn_labels = spectral.labels_
```

- laplacian_kernel: k(x,y) = exp(-$\gamma\|x - y\|_1$),
  - γ=1/features_count

Clustering Results

- K-means: 0.321828

- Aggromerative: 0.379261

- Spectral (laplacian_kernel): 0.343358

- Spectral (nearest neighbors): 0.347362

- Με PCA
  - K-means: 0.371994
  - Agglomerative: 0.221339
  - Spectral (laplacian kernel): 0.339670
  - Spectral (nearest neighbors): 0.401819

```
pca = PCA(n_components=0.90)
pca.fit(x)
```
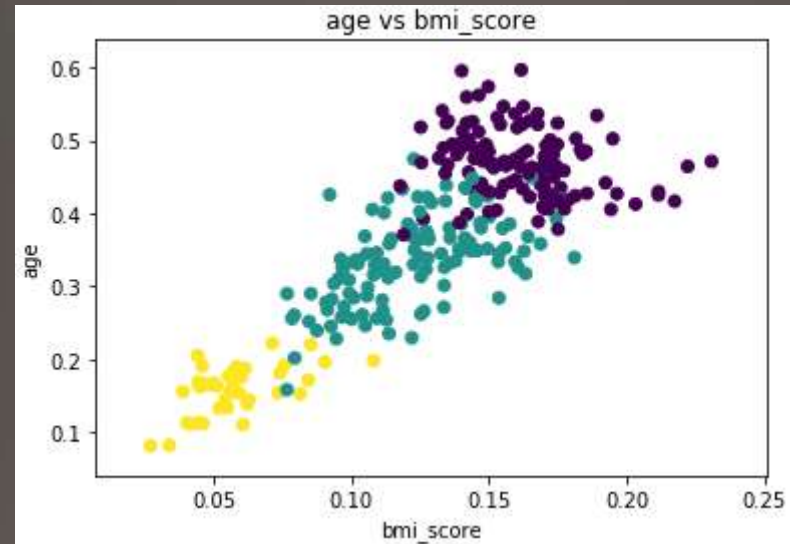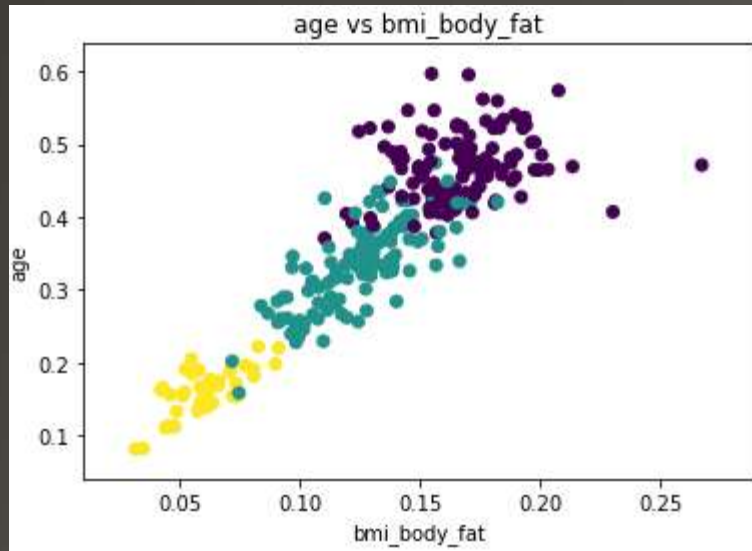
K-means silhouette
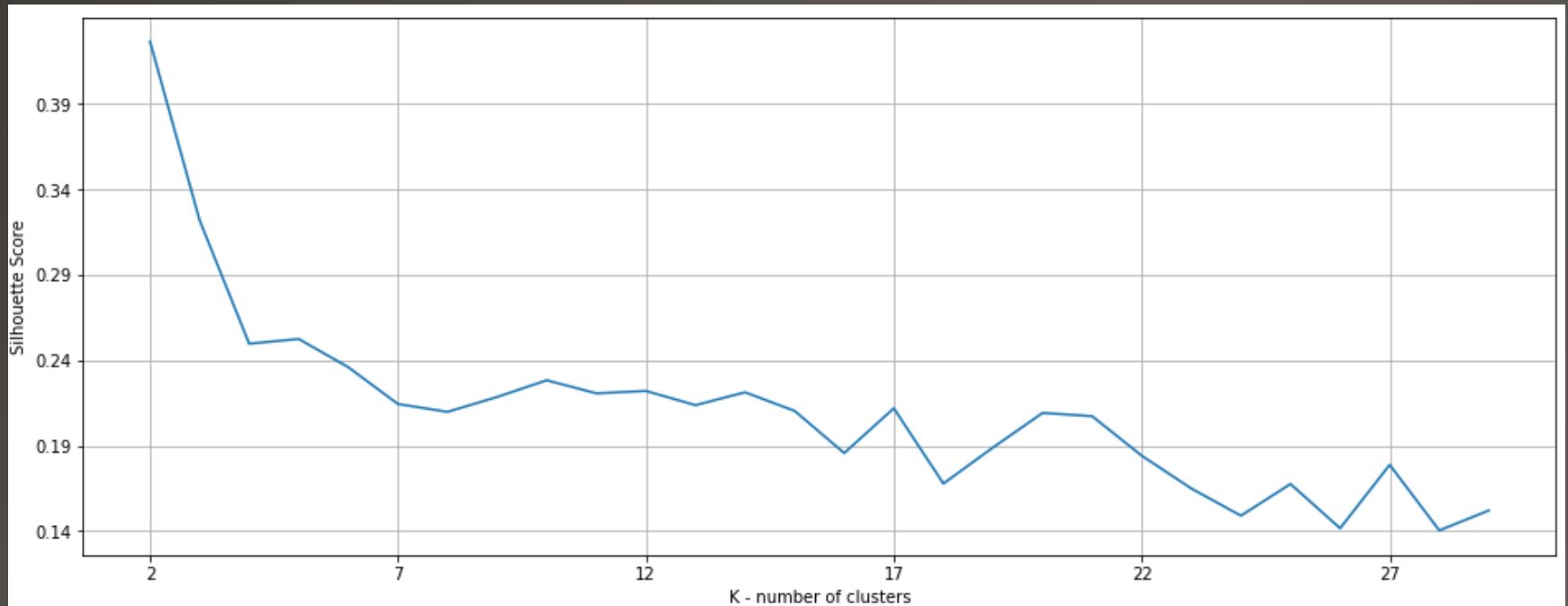
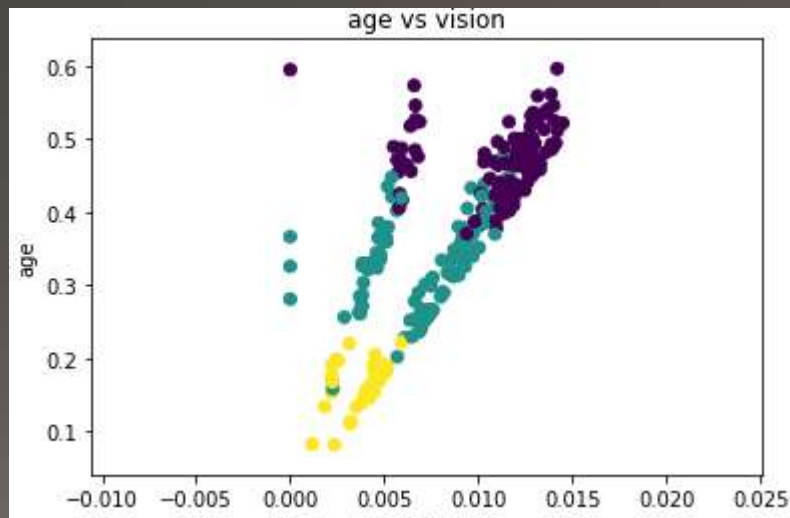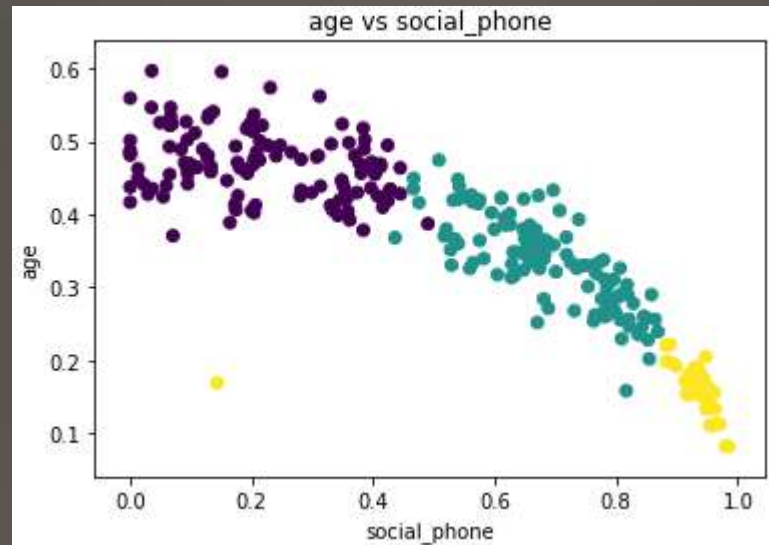# Agglomerative silhouette
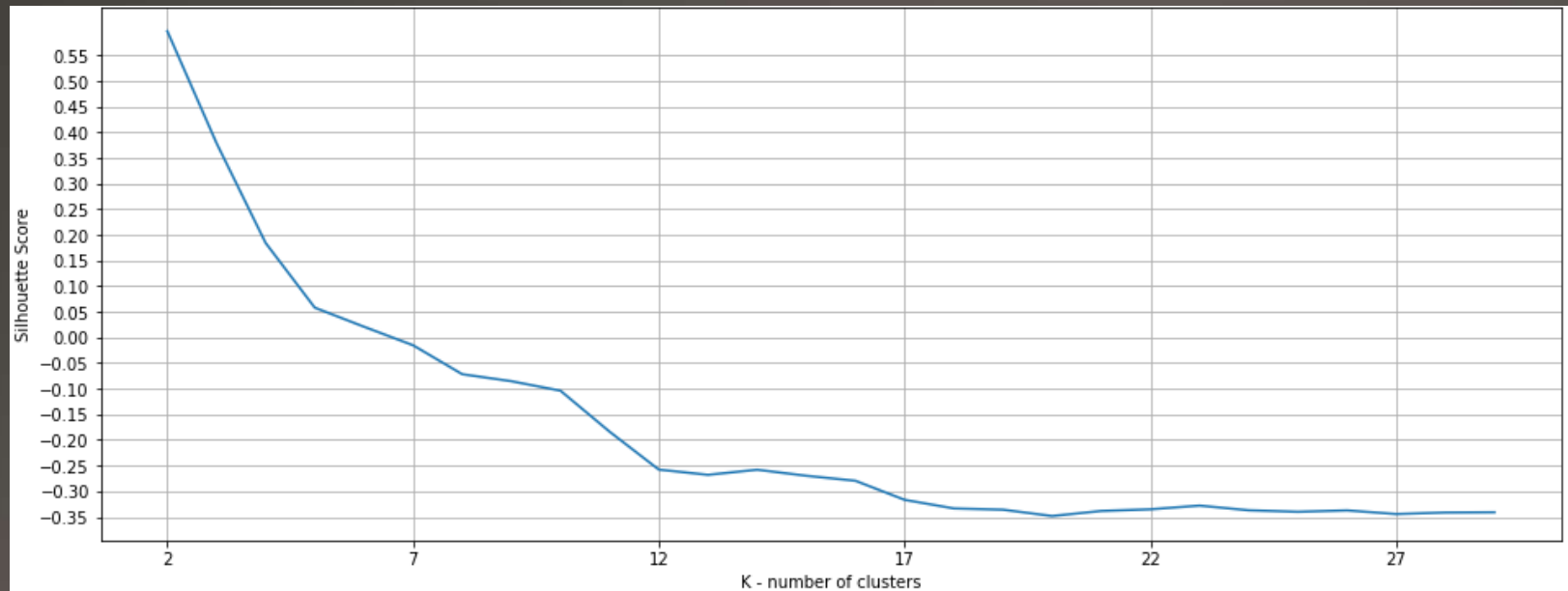
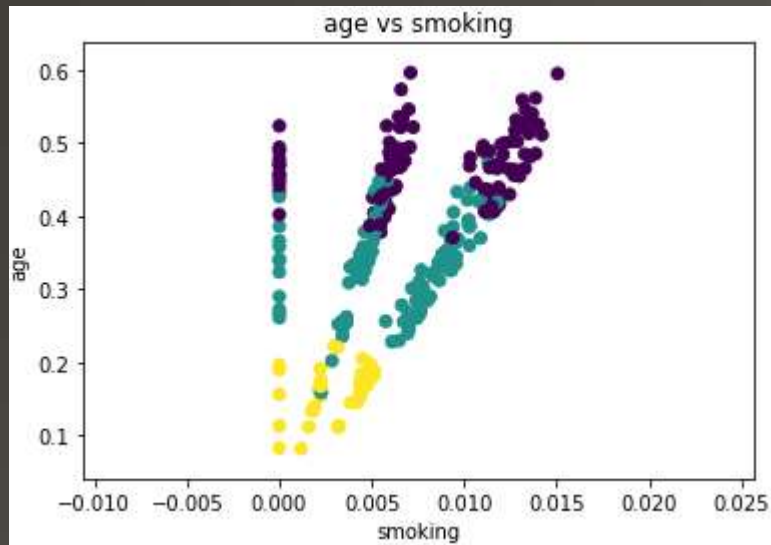# Laplacian kernel silhouette
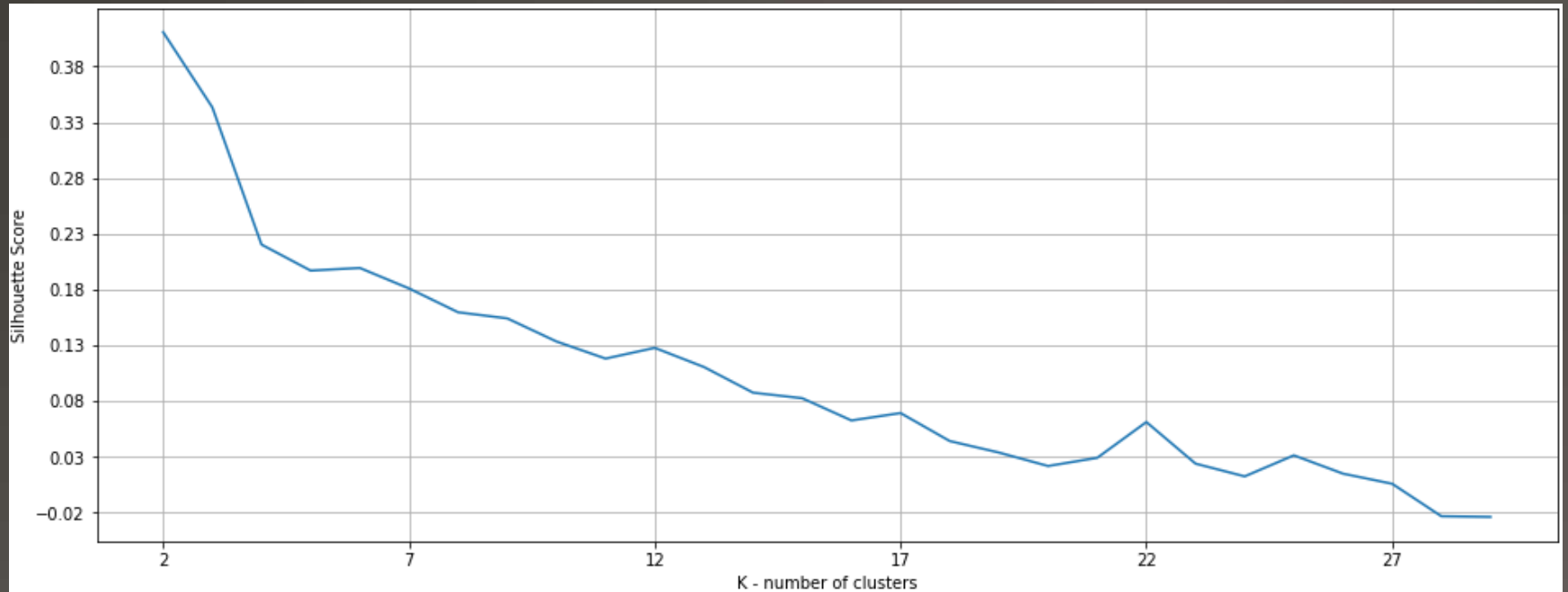
Laplacian NN silhouette
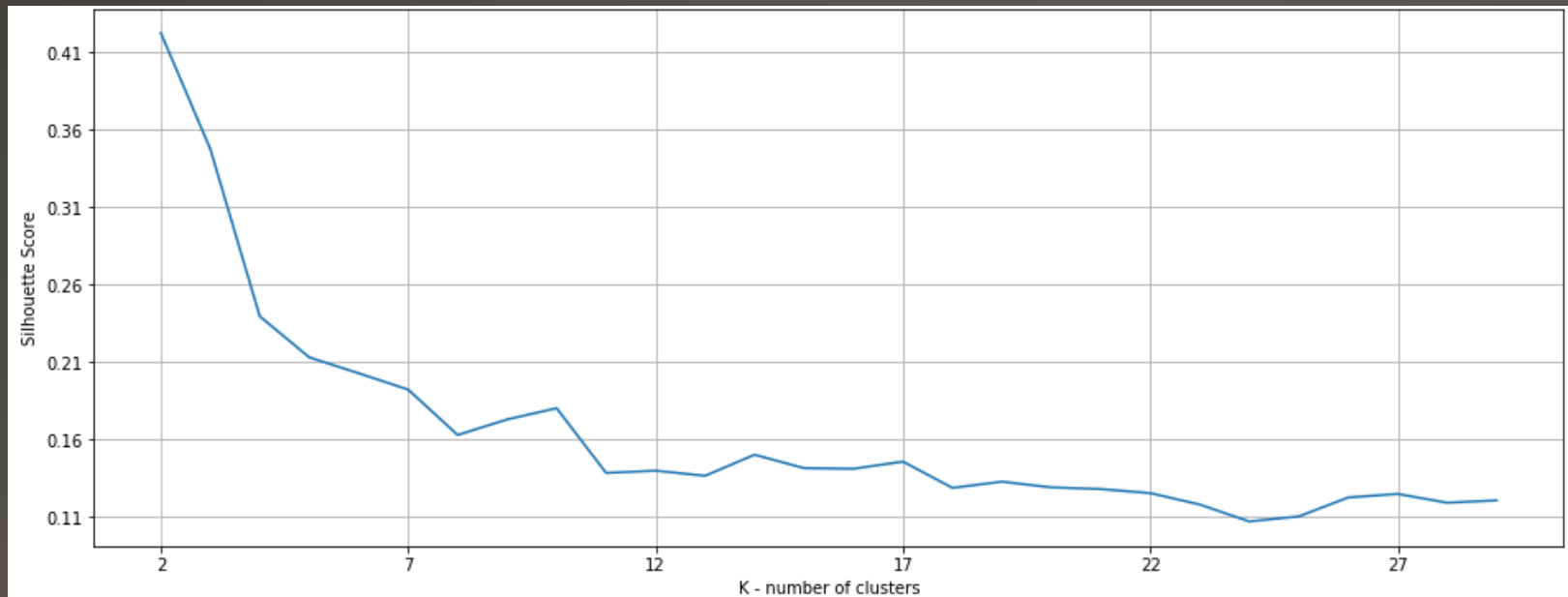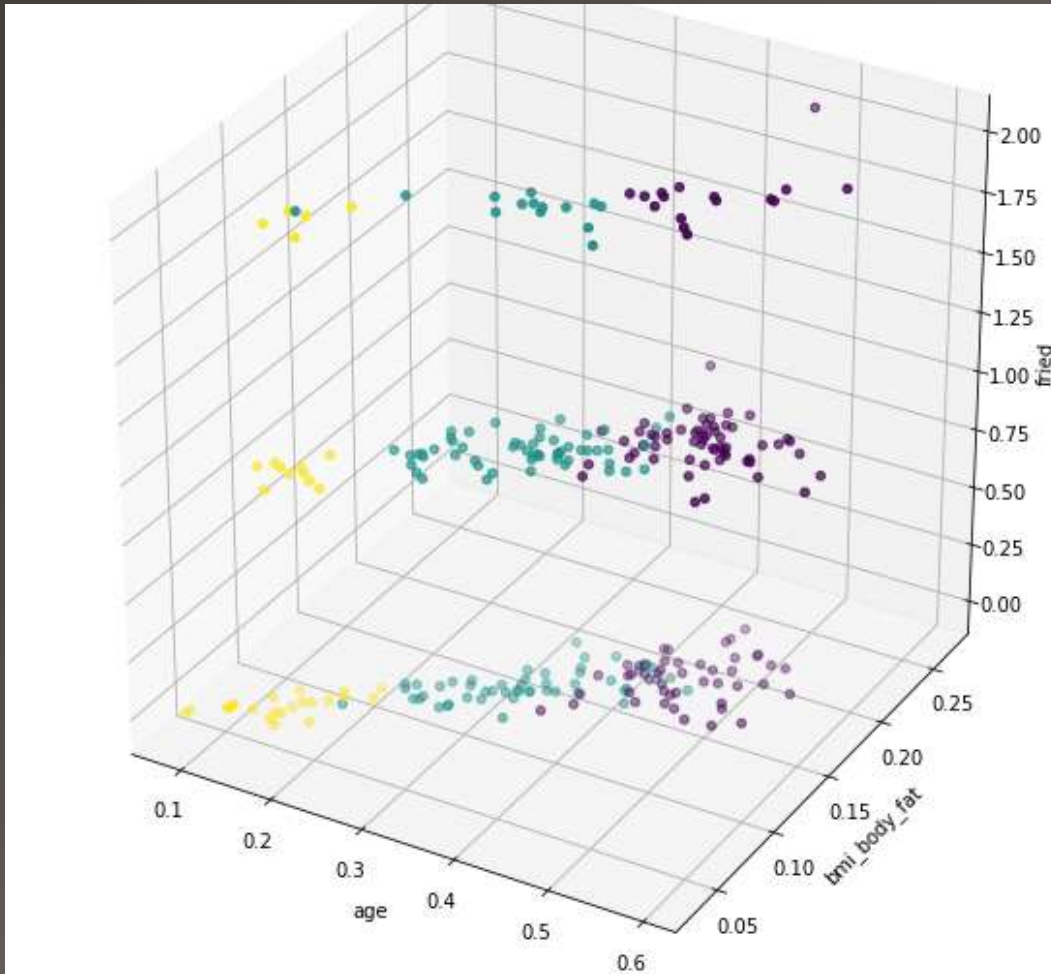
Clustering Visualization (1)
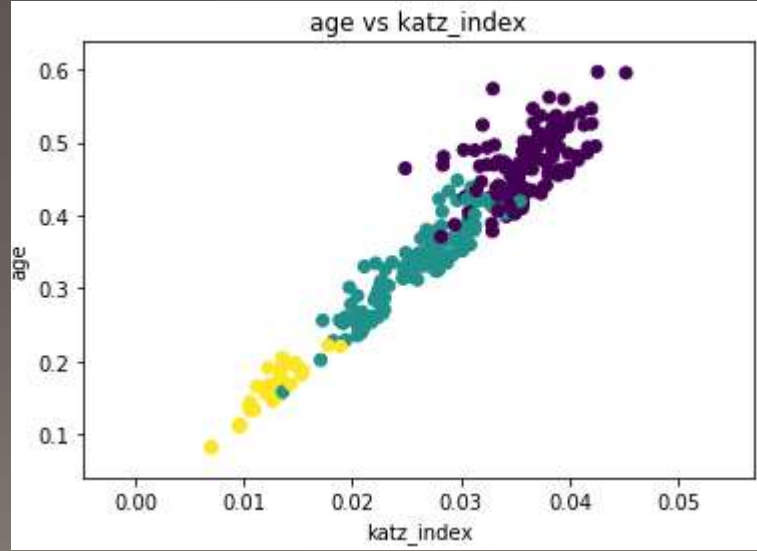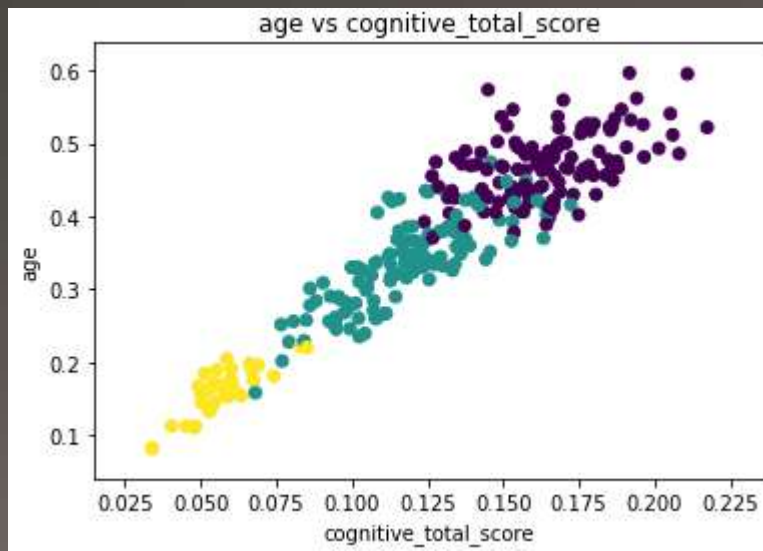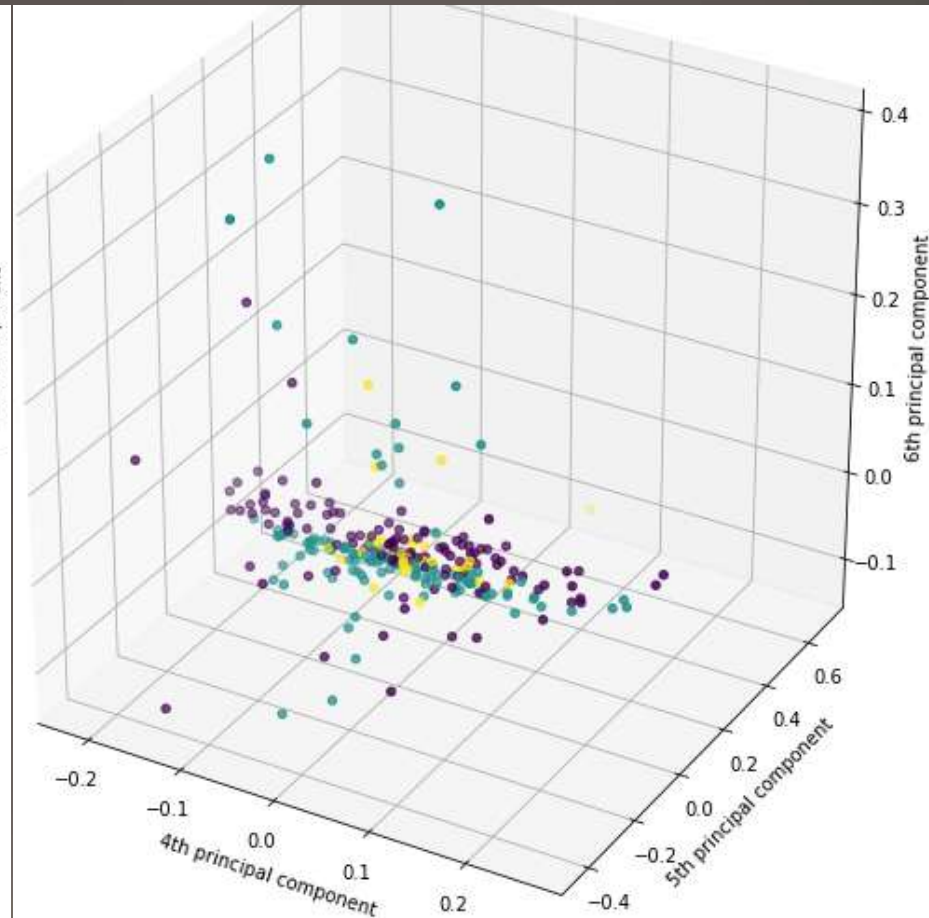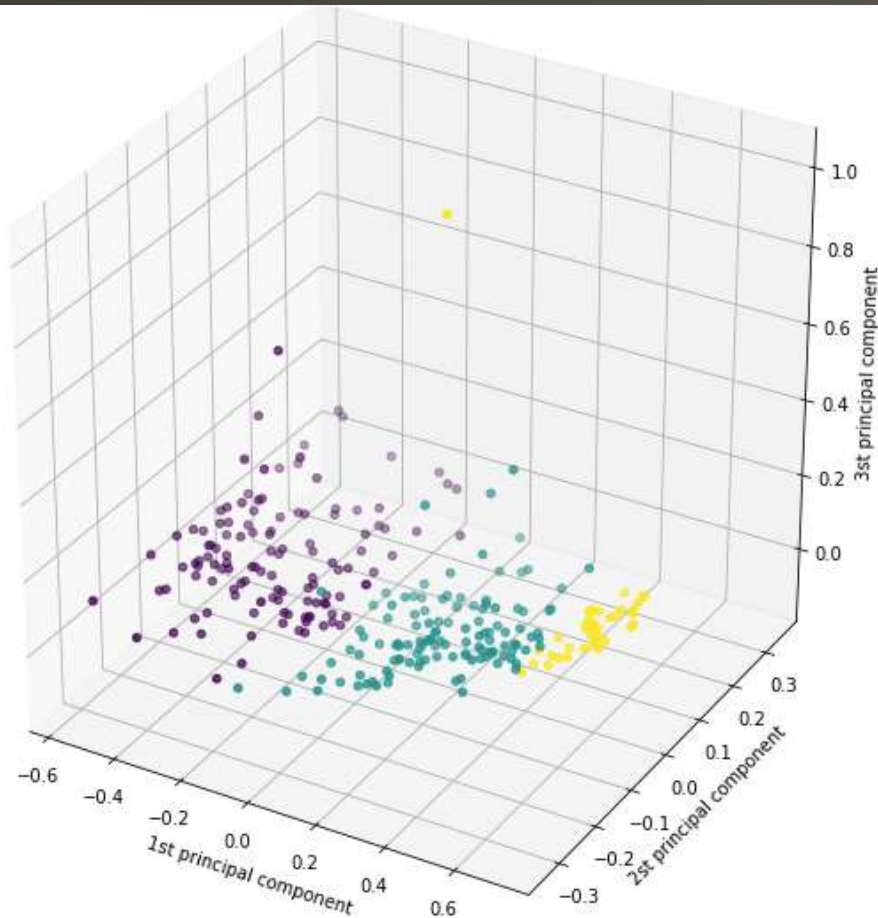
Clustering Visualization (2)
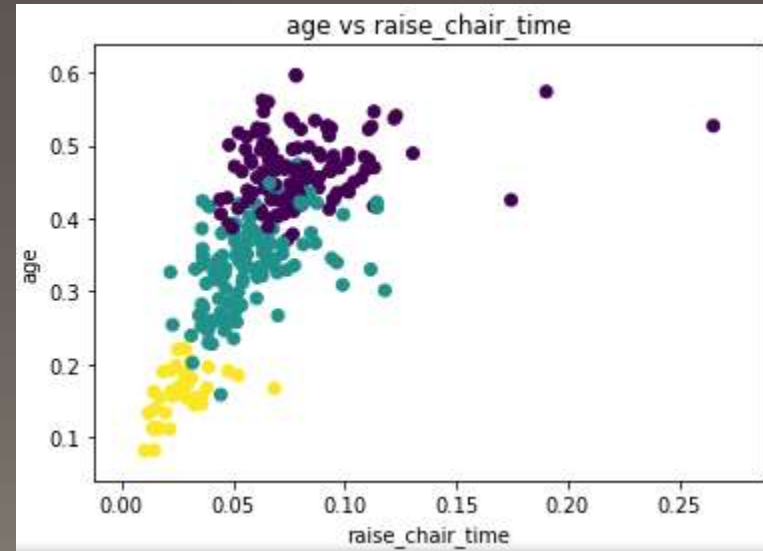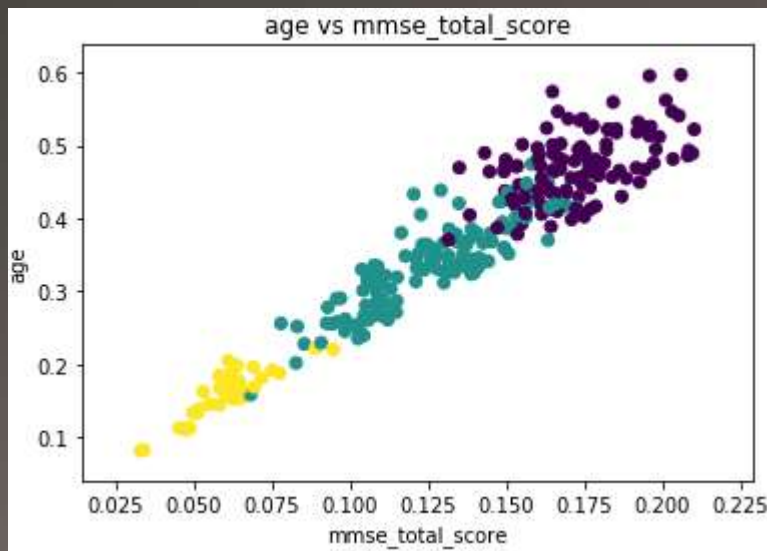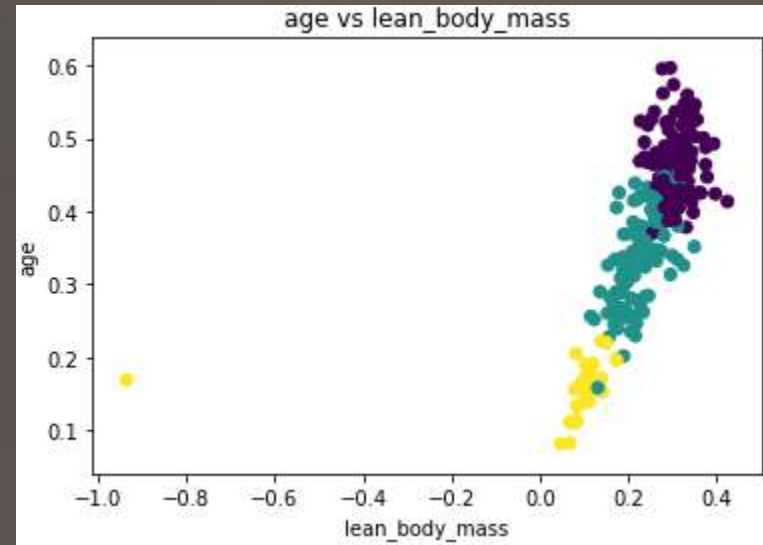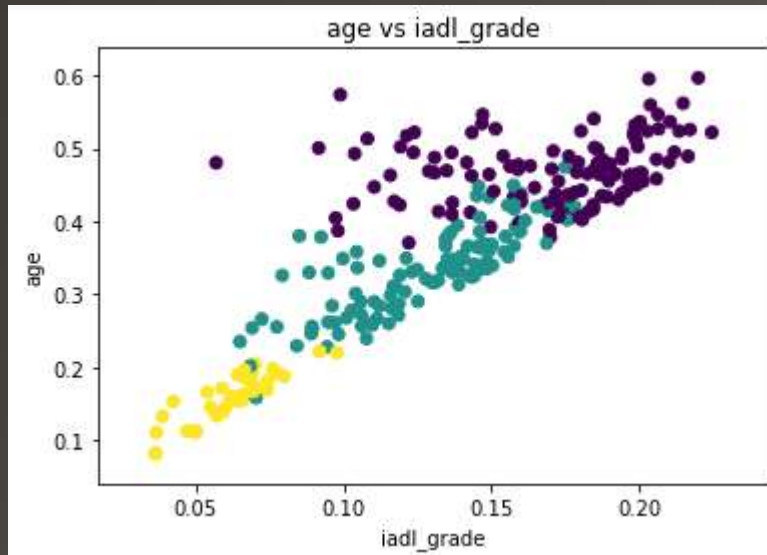
Clustering Visualization (3)
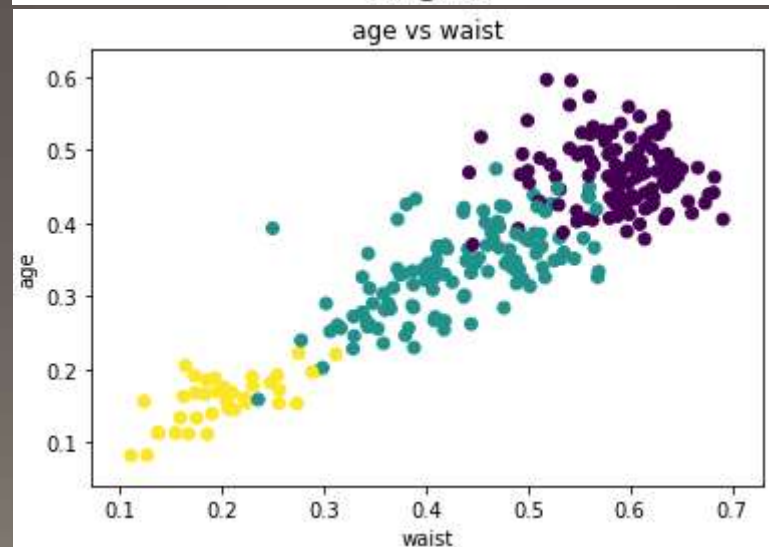
Clustering Visualization (Fried)

PCA Visualization

Ευχαριστώ!