Data Preprocessing

1 Εισαγωγή

1.1 Εισαγωγή στο Data Preprocessing

Η διαδικασία της προεπεξεργασίας δεδομένων αποτελεί ίσως το πιο σημαντικό βήμα στην εξόρυξη δεδομένων. Τα σύνολα δεδομένων που χρησιμοποιούνται μπορεί συχνά να περιέχουν θόρυβο, περιττές πληροφορίες κ.α. ενώ πολλές φορές και η μορφή τους δεν είναι κατάλληλη για την εφαρμογή αλγορίθμων. Η προεπεξεργασία που γίνεται στα δεδομένα αφορά σε πλήθος μεθόδων που αφορούν τον καθαρισμό των δεδομένων (Data Cleansing) (π.χ. Fill in missing values, smooth noisy data, identify or remove outliers and noisy data, and resolve inconsistencies), τη δειγματοληψία τους (Data Sampling), το μετασχηματισμό τους π.χ. κανονικοποίηση (Data Normalization), την επιλογή και την εξαγωγή χαρακτηριστικών (Feature Selection and Extraction) κ.α. Τέλος, όσον αφορά την εξαγωγή χαρακτηριστικών, διακρίνουμε τις μεθόδους σε γραμμικές και μη γραμμικές, όπως PCA και ISOMAP αντίστοιχα.

1.2 Data Preprocessing στην Python

Για αν αφαιρέσουμε τις διπλοεγγραφές από τα δεδομένα εκτελούμε την εντολή:

```
>>> df.drop_duplicates(subset=[columns])
```

Για απλούς μετασχηματισμούς χρησιμοποιούμε τη συνάρτηση StandardScaler. Η StandardScaler έχει τα ορίσματα with_mean και with_std. Με το with_mean = True (default) αφαιρεί από τα δεδομένα το μέσο όρο τους (δηλαδή τα κεντράρει στην αρχή των αξόνων). Με το παράμετρο with_std = True (default) διαιρεί τα δεδομένα με την τυπική τους απόκλιση, όπως παρακάτω:

```
>>> from sklearn.preprocessing import StandardScaler
>>> scaler = StandardScaler()
>>> scaler = scaler.fit(data)
>>> transformed = pd.DataFrame(scaler.transform(data))
```

Νέα δεδομένα μπορούν και αυτά να γίνουν scale, αρκεί να χρησιμοποιηθεί ο ήδη εκπαιδευμένος scaler:

```
>>> scaler.transform(data)
```

Για να κάνουμε normalize ένα διάνυσμα x στο εύρος [0, 1] εκτελούμε την εντολή:

```
>>> normalized_x = [(float(i) - min(x))/(max(x) - min(x)) for i in x]
```

Για να εφαρμόσουμε το normalization σε ένα data frame εκτελούμε την εντολή:

```
>>> normalized_df=(df - df.min())/(df.max() - df.min())
```

Για να διακριτοποιήσουμε τα δεδομένα χρησιμοποιούμε την pd.cut, η οποία λαμβάνει ως όρισμα μια ακολουθία από τα σημεία στα οποία θα γίνει διακριτοποίηση, π.χ. η εντολή:

```
>>> pd.cut(x, bins=10)
```

διακριτοποιεί τα δεδομένα σε 10 διαστήματα.

Για να εφαρμόσουμε δειγματοληψία στα δεδομένα χρησιμοποιούμε την εντολή df.sample. Για να λάβουμε π.χ. 100 εγγραφές εκτελούμε την εντολή:

```
> data_sample = data.sample(n=100, random_state=1, replace=True)
```

όπου με την παράμετρο replace ελέγχουμε αν η δειγματοληψία θα γίνει με αντικατάσταση.

Τέλος, για να υπολογίσουμε το correlation των δεδομένων μπορούμε να χρησιμοποιήσουμε την pd.`corr`, ενώ για το covariance μπορούμε να χρησιμοποιήσουμε την pd.`cov`.

1.2 PCA στην Python

Για να υπολογίζουμε το PCA εκτελούμε τη συνάρτηση sklearn.decomposition.PCA:

```
>>> from sklearn.decomposition import PCA

>>> pca = PCA(n_components=2)

>>> pca = pca.fit(X)

>>> transformed = pca.transform(X)
```

Μπορούμε να λάβουμε τις ιδιοτιμές και τα ιδιοδιανύσματα με τις εντολές:

```
>>> eigenvalues = pca.explained_variance_

>>> eigenvectors = pca.components_
```

1.3 ISOMAP στην Python

Για να υπολογίζουμε το ISOMAP χρησιμοποιούμε τη συνάρτηση sklearn.manifold.Isomap:

```
>>> from sklearn.manifold import Isomap

>>> isomap = Isomap(n_neighbors = 4, n_components = 2)
```

```
>>> isomap = isomap.fit(X)
>>> transformed = isomap.transform(X)
```

όπου με την παράμετρο n_components επιλέγουμε τον αριθμό των διαστάσεων που θα μετασχηματίσουμε τα δεδομένα και με το n_neighbors επιλέγουμε το πλήθος των κοντινότερων αποστάσεων για κάθε σημείο.

2 Προεπεξεργασία Δεδομένων – Χρήση Μεθόδων για Καθαρισμό, Κανονικοποίηση, Διακριτοποίηση και Δειγματοληψία

Για την προεπεξεργασία δεδομένων θα χρησιμοποιήσουμε τις δύο πρώτες στήλες των δεδομένων του αρχείου που δίνεται (engdata.txt).

Ένα summary των δεδομένων είναι το παρακάτω:

```
      Age             Salary
 Min.   :20.00   Min.   : 475
 1st Qu.:44.00   1st Qu.:1156
 Median :51.00   Median :1517
 Mean   :49.94   Mean   :1592
 3rd Qu.:57.00   3rd Qu.:1969
 Max.   :70.00   Max.   :3900
```

Εισάγουμε τα δεδομένα με τις παρακάτω εντολές:

```
>>> import pandas as pd
>>> engdata = pd.read_csv("./engdata.txt")
>>> pdata = engdata.loc[:, ["Age", "Salary"]]
```

Θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Διαγράψτε τις διπλοεγγραφές από τα δεδομένα.

β) Εφαρμόστε τους μετασχηματισμούς center και scale στα δεδομένα και σχεδιάστε τα δεδομένα πριν και μετά από αυτούς τους μετασχηματισμούς.

γ) Εφαρμόστε δειγματοληψία με αντικατάσταση στα δεδομένα επιλέγοντας 150 τιμές. Σχεδιάστε τα δεδομένα πριν και μετά τη δειγματοληψία. Διατηρείται η δομή των δεδομένων;

δ) Εφαρμόστε διακριτοποίηση στα δεδομένα.

2.1 Καθαρισμός και Μετασχηματισμοί Δεδομένων

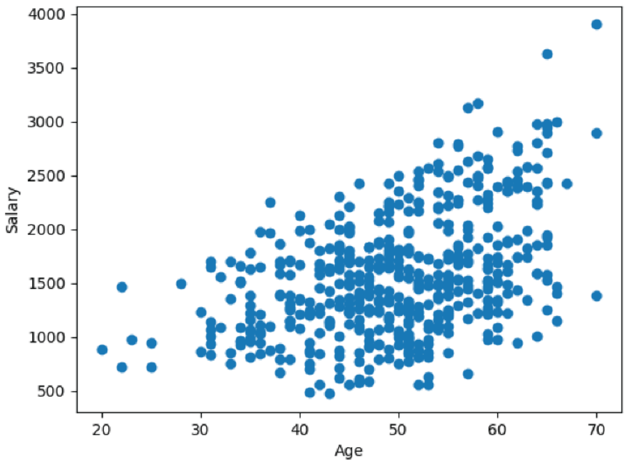
Αρχικά αφαιρούμε τα διπλότυπα από τα δεδομένα (ερώτημα (α)) με την παρακάτω εντολή:

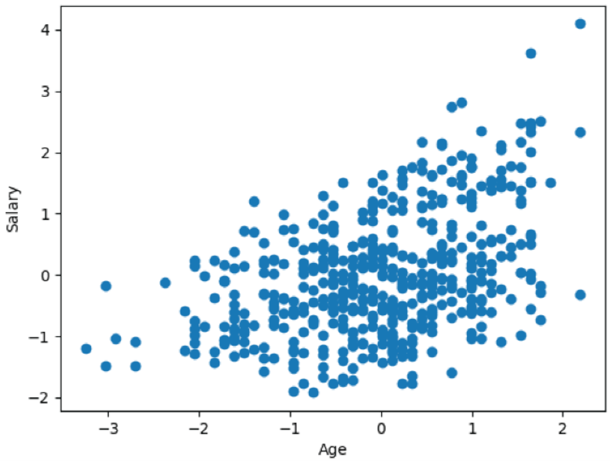
```
>>> pdata = pdata.drop_duplicates()
```

Στη συνέχεια, αφαιρούμε το μέσο όρο και διαιρούμε με την τυπική απόκλιση (ερώτημα (β)):

```
>>> from sklearn.preprocessing import StandardScaler

>>> import matplotlib.pyplot as plt

>>> scaler = StandardScaler()

>>> scaler = scaler.fit(pdata)

>>> transformed = pd.DataFrame(scaler.transform(pdata),
columns=["Age", "Salary"])
```

Μπορούμε να σχεδιάσουμε τα δεδομένα πριν και μετά τους μετασχηματισμούς με τις εντολές `plt.scatter(pdata)` και `plt.scatter(transformed)`.



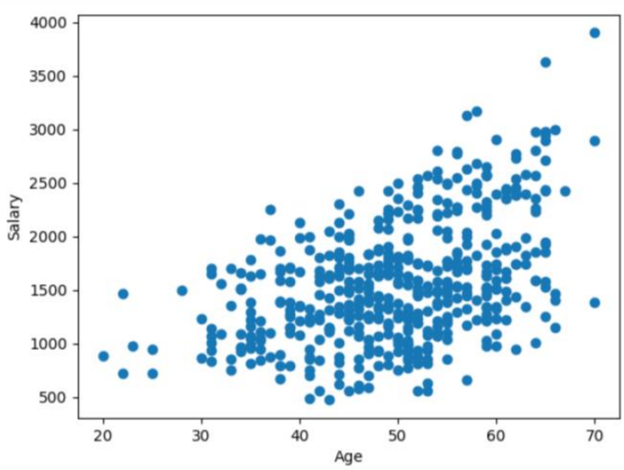


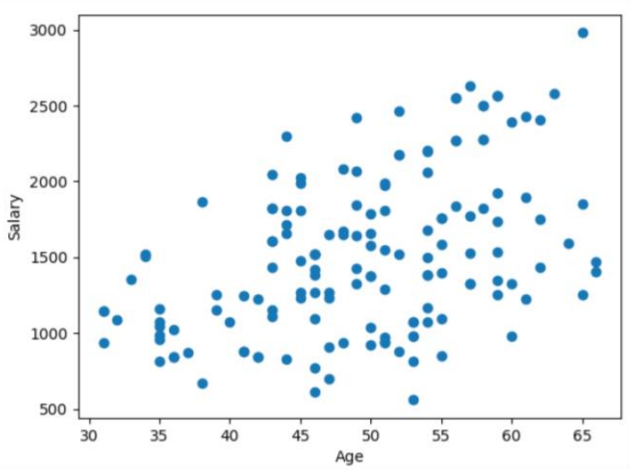
2.2 Δειγματοληψία Δεδομένων

Η δειγματοληψία (ερώτημα (γ)) γίνεται με την εντολή sample:

```
>>> data_sample = pdata.sample(n=150, random_state=1, replace=True)
```

Μπορούμε να σχεδιάσουμε τα δεδομένα πριν και μετά τους μετασχηματισμούς με τις εντολές `plt.scatter(pdata)` και `plt.scatter(sampdata)`.





2.3 Διακριτοποίηση Δεδομένων

Η διακριτοποίηση (ερώτημα (δ)) γίνεται με την εντολή cut. Για τις μεταβλητές Age και Salary επιλέγουμε κατάλληλα bins με τις παρακάτω εντολές:

```
>>> discAge = pd.cut(pdata.Age, [0, 10, 20, 30, 40, 50, 60, 70, 80])
```

```
>>> discSalary = pd.cut(pdata.Salary, pd.interval_range(start=0,
freq=400, end=4000))
```

3 Εξαγωγή Χαρακτηριστικών με PCA

Για το μετασχηματισμό ενός συνόλου δεδομένων με PCA θα χρησιμοποιήσουμε ως εφαρμογή τα δεδομένα του παρακάτω πίνακα.

	X	**Y**	**Z**
X1	1	0	-1
X2	0	1	-1
X3	-1	1	0
X4	0	-1	1
X5	-1	0	1
X6	1	-1	0

Θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Σχεδιάστε τα δεδομένα σε 3 διαστάσεις.

β) Να εφαρμοστεί ο αλγόριθμος PCA για τα διανύσματα x1 – x6 και να αντικατασταθούν από διανύσματα 2 διαστάσεων κατά τέτοιοι τρόπο ώστε η απώλεια πληροφορίας να είναι ελάχιστη.

γ) Επαναλάβετε το ερώτημα (α) χρησιμοποιώντας την Python.

δ) Σχεδιάστε τα νέα δεδομένα μετά το μετασχηματισμό.

3.1 Κατασκευή Δεδομένων και Εισαγωγή Βιβλιοθηκών

Αρχικά κατασκευάζουμε τα δεδομένα:

```
>>> X = [1, 0, -1, 0, -1, 1]

>>> Y = [0, 1, 1, -1, 0, -1]

>>> Z = [-1, -1, 0, 1, 1, 0]

>>> labels = ["x1", "x2", "x3", "x4", "x5", "x6"]

>>> pdata = pd.DataFrame({"X": X, "Y": Y, "Z": Z}, index=labels)
```

3.2 Εφαρμογή Αλγορίθμου PCA

Για το ερώτημα (α) εκτελούμε:

```
>>> fig = plt.figure()

>>> ax = fig.add_subplot(projection='3d')

>>> ax.scatter(pdata.X, pdata.Y, pdata.Z)

>>> for i in range(len(pdata.index)):

>>>     ax.text(pdata.loc[labels[i], "X"], pdata.loc[labels[i], "Y"],
pdata.loc[labels[i], "Z"], '%s' % (str(labels[i])), size=20,
zorder=1)

>>> ax.set_xlabel('X Axis')

>>> ax.set_ylabel('Y Axis')

>>> ax.set_zlabel('Z Axis')

>>> plt.show()
```

Για το ερώτημα (β), αρχικά ορίζουμε τους πίνακες X και $X^T X$:

$$X = [x_1 \quad x_2 \quad \dots \quad x_n]^T = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 1 & 0 & -1 & 0 & -1 & 1 \\ 0 & 1 & 1 & -1 & 0 & -1 \\ -1 & -1 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & -2 & -2 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{bmatrix}$$

Στην συνέχεια απαιτείται ο υπολογισμός των ιδιοτιμών και των ιδιοδιανυσμάτων του πίνακα X^TX. Οι ιδιοτιμές του πίνακα X^TX είναι οι ρίζες του χαρακτηριστικού του πολυωνύμου. Ισχύει:

$$|X^T X - \lambda I_3| = 0 \Rightarrow \begin{vmatrix} 4-\lambda & -2 & -2 \\ -2 & 4-\lambda & -2 \\ -2 & -2 & 4-\lambda \end{vmatrix} = 0 \Rightarrow (\lambda - 6) \cdot \lambda \cdot (6 - \lambda) = 0$$

Άρα οι λύσεις του χαρακτηριστικού πολυωνύμου είναι $\lambda_{1,2} = 6, \lambda_3 = 0$

Επόμενο βήμα αποτελεί η εύρεση των ιδιοδιανυσμάτων που αντιστοιχούν στις 2 μεγαλύτερες ιδιοτιμές αφού ζητείται η αναπαράσταση να γίνει στις 2 διαστάσεις. Για το 1ο ιδιοδιάνυσμα $\underline{u_1}$ ισχύει:

$$(X^T X - \lambda_1 I_3)\underline{u_1} = 0 \Rightarrow \begin{bmatrix} -2 & -2 & -2 \\ -2 & -2 & -2 \\ -2 & -2 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0 \Rightarrow x + y + z = 0 \Rightarrow z = -x - y \qquad (1)$$

Η παραπάνω σχέση αποτελεί συνθήκη που πρέπει να ισχύει λόγω της 1ης ιδιοτιμής λ_1, για το 1ο ιδιοδιάνυσμα. Οπότε προκειμένου να υπολογίσουμε το ιδιοδιάνυσμα που αντιστοιχεί στην ιδιοτιμή λ_1 έχουμε:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \overset{(4)}{\Rightarrow} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} - (x + y) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} = x \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

Η παραπάνω σχέση μας δίνει μια οικογένεια ιδιοδιανυσμάτων για τις διάφορες τιμές των x, y που όμως δεν είναι όλα πάντα γραμμικώς ανεξάρτητα και δεν έχουν μέτρο όσο με 1. Έστω επιλέγω ως $\underline{u_1}$ το ακόλουθο διάνυσμα:

$$\underline{u_1} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \qquad (2)$$

Για το 2ο ιδιοδιάνυσμα που επίσης αντιστοιχεί στην ιδιοτιμή $\lambda_1 = \lambda_2 = 6$ πρέπει να ισχύει:

$$\underline{u_1} \perp \underline{u_2} \Rightarrow \underline{u_2}^T \underline{u_1} = 0 \Rightarrow \frac{1}{\sqrt{2}} [x \quad y \quad z] \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = 0 \Rightarrow x - z = 0 \Rightarrow x = z \qquad (3)$$

Με συνδυασμό των σχέσεων (1) και (3) προκύπτει ότι για το $\underline{u_2}$ πρέπει να ισχύει:

$$\xRightarrow{\alpha\pi\acute{o}\ (1) \kappa\alpha\iota (3)} y = -2x$$

Οπότε τελικά έχουμε:

$$\underline{u_2} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ -2x \\ x \end{bmatrix} \xRightarrow{x=1} \underline{u_2} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \xRightarrow[\delta\iota\acute{\alpha}\nu\upsilon\sigma\mu\alpha]{\mu o\nu\alpha\delta\iota\alpha\acute{\iota}o} \underline{u_2} = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \quad (4)$$

Έχοντας υπολογίσει τα 2 ιδιοδιανύσματα που αντιστοιχούν στις 2 μεγαλύτερες ιδιοτιμές (σχέσεις (2) και (4)), η αναγωγή των 3D διανυσμάτων x_i σε αντίστοιχα 2D με την ελάχιστη απώλεια πληροφορίας πραγματοποιείται βάση της σχέσης:

$$x_i' = \begin{bmatrix} \underline{u_1^T} x_i \\ \underline{u_2^T} x_i \end{bmatrix}$$

Έτσι προκύπτουν τα νέα δισδιάστατα διανύσματα.

$$x_1' = \begin{bmatrix} \frac{2}{\sqrt{2}} \\ 0 \end{bmatrix}, x_2' = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{3}{\sqrt{6}} \end{bmatrix}, x_3' = \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ -\frac{3}{\sqrt{6}} \end{bmatrix}, x_4' = \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{3}{\sqrt{6}} \end{bmatrix}, x_5' = \begin{bmatrix} -2\sqrt{2} \\ 0 \end{bmatrix}, x_6' = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{3}{\sqrt{6}} \end{bmatrix}$$

Η ποσότητα της πληροφορίας που χάνεται με τη νέα αυτή αναπαράσταση σε σχέση με αυτήν της εκφώνησης είναι βάση της θεωρίας:

$$\alpha\pi\acute{\omega}\lambda\varepsilon\iota\alpha\ \pi\lambda\eta\rho o\varphi o\rho\acute{\iota}\alpha\varsigma \Rightarrow \frac{\sum(\iota\delta\iota o\tau\iota\mu\acute{\omega}\nu\ \pi o\upsilon\ \delta\varepsilon\ \chi\rho\eta\sigma\iota\mu o\pi o\iota o\acute{\upsilon}\nu\tau\alpha\iota)}{\sum(\acute{o}\lambda\omega\nu\ \tau\omega\nu\ \iota\delta\iota o\tau\iota\mu\acute{\omega}\nu)} = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} = 0$$

Επομένως δεν υπάρχει καμιά απώλεια πληροφορίας από την μετάβαση από τις τρεις διαστάσεις στις 2.

4 Επιλογή και Εξαγωγή Χαρακτηριστικών (Correlation, PCA)

Για την κατασκευή ενός μοντέλου PCA θα χρησιμοποιήσουμε ως εφαρμογή τα δεδομένα εκπαίδευσης του αρχείου που δίνεται (engdata.txt) για ένα πρόβλημα δυαδικής ταξινόμησης.

Ένα summary των δεδομένων είναι το παρακάτω:

```
     Age            Salary        YearsOfStudy        WorkExp      Location
Min.   :20.00   Min.   : 475   Min.   : 3.000   Min.   :10.0   EU:345
1st Qu.:44.00   1st Qu.:1156   1st Qu.: 5.000   1st Qu.:39.0   US:305
Median :51.00   Median :1517   Median : 6.000   Median :46.0
Mean   :49.94   Mean   :1592   Mean   : 6.032   Mean   :44.9
3rd Qu.:57.00   3rd Qu.:1969   3rd Qu.: 7.000   3rd Qu.:52.0
Max.   :70.00   Max.   :3900   Max.   :10.000   Max.   :68.0
```

Αρχικά, εισάγουμε τα δεδομένα και τα διαχωρίζουμε σε χαρακτηριστικά και χαρακτηριστικό κλάσης:

```
>>> engdata = pd.read_csv("./engdata.txt")

>>> location = engdata.Location

>>> engdata = engdata.drop(["Location"], axis=1)
```

Στη συνέχεια, θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Σχεδιάστε τις τιμές του συνόλου δεδομένων ανά δύο χαρακτηριστικά με διαφορετικά χρώματα/σύμβολα για κάθε κλάση.

β) Υπολογίστε το correlation matrix για τα χαρακτηριστικά. Ποια χαρακτηριστικά θα μπορούσαν να παραληφθούν;

γ) Κατασκευάστε ένα μοντέλο PCA από τα δεδομένα και βρείτε τις ιδιοτιμές και τα ιδιοδιανύσματα. Σχεδιάστε το ποσοστό πληροφορίας για τα principal components.

δ) Εφαρμόστε το μοντέλο στα δεδομένα και διατηρήστε τις δύο πρώτες διαστάσεις. Σχεδιάστε τα νέα δεδομένα σε δισδιάστατο άξονα.

ε) Ανακατασκευάστε τα δεδομένα και σχεδιάστε τα εκ νέου (όπως στο (α)). Υπολογίστε την απώλεια της πληροφορίας.

4.1 Σχεδίαση Δεδομένων και Υπολογισμός Correlation

Για τα ερωτήματα (α) και (β) εκτελούμε τις εντολές:

```
>>> plt.scatter(engdata[(location == "EU")].Age, engdata[(location ==
"EU")].Salary, c="red", marker="+")

>>> plt.scatter(engdata[(location == "US")].Age, engdata[(location ==
"US")].Salary, c="blue", marker="o")

>>> plt.show()

>>> print(engdata.corr())
```

4.2 Εφαρμογή Αλγορίθμου PCA

Για το (γ) υπολογίζουμε αρχικά το PCA και βρίσκουμε ιδιοτιμές και ιδιοδιανύσματα:

```
>>> from sklearn.decomposition import PCA

>>> scaler = StandardScaler()

>>> scaler = scaler.fit(engdata)

>>> transformed = pd.DataFrame(scaler.transform(engdata),
columns=engdata.columns)

>>> pca = PCA()

>>> pca = pca.fit(transformed)

>>> pca_transformed = pca.transform(transformed)

>>> eigenvalues = pca.explained_variance_

>>> eigenvectors = pca.components_
```

Για να σχεδιάσουμε το ποσοστό πληροφορίας για τα principal components εκτελούμε την παρακάτω εντολή:

```
>>> plt.bar(range(len(eigenvalues)), eigenvalues/sum(eigenvalues))

>>> plt.show()
```

Εφαρμόζουμε το PCA στα δεδομένα ώστε να μειώσουμε τις διαστάσεις τους σε δύο και στη συνέχεια τα σχεδιάζουμε (ερώτημα (δ)):

```
>>> pca = PCA(n_components=2)

>>> pca_transformed = pd.DataFrame(pca.fit_transform(transformed))

>>> plt.scatter(pca_transformed.loc[:, 0], pca_transformed.loc[:, 1])

>>> plt.show()
```

Τέλος, για να ανακατασκευάσουμε τα δεδομένα (ερώτημα (ε)) και να τα σχεδιάσουμε εκτελούμε τις εντολές:

```
>>> pca_inverse =
pd.DataFrame(pca.inverse_transform(pca_transformed),
columns=engdata.columns)

>>> plt.scatter(pca_inverse[(location == "EU")].Age,
pca_inverse[(location == "EU")].Salary, c="red", marker="+")

>>> plt.scatter(pca_inverse[(location == "US")].Age,
pca_inverse[(location == "US")].Salary, c="blue", marker="o")

>>> plt.show()
```

Η απώλεια της πληροφορίας υπολογίζεται με την εντολή:

```
>>> info_loss = (eigenvalues[2] + eigenvalues[3]) / sum(eigenvalues)

>>> print(info_loss)
```

5 Μη Γραμμική Εξαγωγή Χαρακτηριστικών (ISOMAP)

Για την κατασκευή ενός μοντέλου ISOMAP θα χρησιμοποιήσουμε ως εφαρμογή τα δεδομένα εκπαίδευσης του αρχείου που δίνεται (srdata.txt).

Ένα summary των δεδομένων είναι το παρακάτω:

```
       V1                   V2                   V3
 Min.   :-0.4738629   Min.   :-0.55193   Min.   :0.0006573
 1st Qu.:-0.1502609   1st Qu.:-0.21918   1st Qu.:0.1290643
 Median :-0.0006371   Median :-0.02540   Median :0.2416298
 Mean   : 0.0058976   Mean   :-0.05905   Mean   :0.2498458
 3rd Qu.: 0.1935721   3rd Qu.: 0.08422   3rd Qu.:0.3734662
 Max.   : 0.6282747   Max.   : 0.39584   Max.   :0.4999653
```

Αρχικά, εισάγουμε τα δεδομένα:

```
>>> srdata = pd.read_csv("./srdata.txt")
```

Στη συνέχεια, θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Σχεδιάστε τα δεδομένα σε τρισδιάστατο γράφημα.

β) Εφαρμόστε τον αλγόριθμο ISOMAP με k = 4 ώστε να μετασχηματίσετε τα δεδομένα σε δύο διαστάσεις.

γ) Σχεδιάστε εκ νέου τα δεδομένα σε τρισδιάστατο γράφημα με διαφορετικά χρώματα ανάλογα με το αποτέλεσμα του ISOMAP (την πρώτη στήλη).

δ) Σχεδιάστε τα νέα δεδομένα σε ένα νέο δισδιάστατο γράφημα, με διαφορετικά χρώματα ανάλογα με το αποτέλεσμα του ISOMAP (την πρώτη στήλη).

5.1 Σχεδίαση Δεδομένων

Για το ερώτημα (α) εκτελούμε τις εντολές:

```
>>> fig = plt.figure()

>>> ax = fig.add_subplot(projection='3d')

>>> ax.scatter(srdata.V1, srdata.V2, srdata.V3)

>>> ax.set_xlabel('V1')

>>> ax.set_ylabel('V2')

>>> ax.set_zlabel('V3')

>>> plt.show()
```

οπότε προκύπτει το σχήμα που ζητείται.

5.2 Εφαρμογή Αλγορίθμου ISOMAP

Για το (β) υπολογίζουμε το ISOMAP και λαμβάνουμε τα νέα δεδομένα:

```
>>> from sklearn.manifold import Isomap

>>> isomap = Isomap(n_neighbors = 4, n_components = 2)

>>> isomap = isomap.fit(srdata)

>>> transformed = pd.DataFrame(isomap.transform(srdata))
```

Για να χρησιμοποιήσουμε το αποτέλεσμα του ISOMAP ως το χρώμα των δεδομένων (ερωτήματα (γ), (δ)), αρχικά κατασκευάζουμε την παρακάτω μεταβλητή:

```
>>> colors = [i - min(transformed.loc[:, 0].tolist()) + 1 for i in
transformed.loc[:, 0].tolist()]
```

Στη συνέχεια εκτελούμε τις παρακάτω εντολές ώστε να σχεδιάσουμε με το χρωματισμό του ISOMAP τα αρχικά δεδομένα (ερώτημα (γ)):

```
>>> fig = plt.figure()

>>> ax = fig.add_subplot(projection='3d')

>>> ax.scatter(srdata.V1, srdata.V2, srdata.V3, c=colors)

>>> ax.set_xlabel('V1')

>>> ax.set_ylabel('V2')

>>> ax.set_zlabel('V3')

>>> plt.show()
```

και τα μετασχηματισμένα δεδομένα σε νέο γράφημα (ερώτημα (δ)):

```
>>> plt.scatter(transformed.loc[:, 0], transformed.loc[:, 1],
c=colors)

>>> plt.show()
```