Uppsala University

Data Mining I

1DL360

Assignment 1 (Python version)

Data Preprocessing (Python)

The objectives of this assignment are:

1. To practice the design of a basic knowledge discovery process.
2. To try different types of data pre-processing.

Preliminary of Language/Software: You can choose Pycharm or Jupyter Notebook (or others that you have been familiar with) as your Python IDE. To install packages, use command `pip install [package name]` in the Windows' or Mac's terminal.

TASK 1: reading the data

1.1

Read the iris_data.csv and iris_labels.csv files via python and preprocess it. These files can be found on Studium. The Iris dataset includes information about three iris species known as Setosa, Versicolor and Virginica. Each instance includes some information about the shape of the flower including sepal length, sepal width, petal length and petal width. As a result, our dataset includes 150 data points/record that have 4 features/attribute and each input is associated with a class, i.e., Setosa, Versicolor, Virginica. The first 4 features are in the iris_data.csv file and the label of each record can be found in the iris_labels.csv file

Import pandas as pd and numpy as np packages with the import command and use pd.read_csv() to read the files.

```
import pandas as pd
import numpy as np

data = pd.read_csv('iris_data.csv')
labels = pd.read_csv('iris_labels.csv')
```

Check the size of the data tables and print the first 5 rows with print(data.shape) and data.head().

```
print(data.shape)
print(data.head())
```

1.2

Join the two csv files into a single data frame.

```
data = pd.merge(data, labels, on='id', how='inner')
```

1.3

Filter out the attribute "examiner", that is not needed for the analysis.

```
data.drop(['examiner'], axis=1, inplace=True)
```

1.4

Order the data frame by the "species" attribute.

```
data = data.sort_values('species')
```

1.5

You can use the seaborn package to see the scatter plot of the dataset.

```
import seaborn as sns
sns.pairplot(data, hue="species")
```

TASK 2: database preprocessing

2.1

Calculate the number of instances in each class (Setosa, Versicolor and Vriginica).

```
print(data.value_counts("species"))
```

You can also use `data.groupby("species").count()`.

TASK 3: data cleaning

During the previous task you should have noticed that some observations have suspicious values, that we should take care of.

3.1

Missing values have been coded using -9999 in the input file. You can filter out the observations containing missing values.

```
data = data[data["attribute_name"] != -9999]
```

3.2

Use again the scatterplots to see if there are outliers. If there are, remove them. If you want to automate this you could use `scipy.stats.zscores` which will tell you how many standard deviations away from the mean a single observation is, for a given variable.

TASK 4: data transformation

The objective of this task is to transform the features describing the objects. We start making the scale of the numerical attributes uniform.

4.1

Perform a min/max normalization on the range [0,1].

```
from sklearn.preprocessing import MinMaxScaler
minmax_scaled = MinMaxScaler().fit_transform(data)
```

4.2

Perform a standardization (in some references it is called z-transformation).

```
from sklearn.preprocessing import StandardScaler
sd_scaled = StandardScaler().fit_transform(data)
```

4.3

Apply PCA to the current data and view the scatterplot of the new data set.

```
from sklearn.decomposition import PCA
pca = PCA()
principal_components = pca.fit_transform(data)
```

Use the attribute `pca.explained_variance_ratio_` to see how many components are retained if one wishes to explain at least 95% of the variance.

4.4

Check the contribution of each attribute on the resulting components.

```
pd.DataFrame(pca.components_, columns=["Sepal L", "Sepal W",
    "Petal L", "Petal W"],
             index=['PC 1','PC 2','PC 3','PC 4']).abs().mean(
    axis=0)
```

4.5

Do the same as in 4.4, but applied to a modified dataset where you rescale the pl attribute to the range [0, 100]. Inspect the definition of the new components. You can use the min-max scaler for this.

4.6

Do the same as in 4.4, but applied to a modified dataset where you add an outlier (noise). For example, modify the value of attribute pl in the first record to 5000. Inspect the definition of the new components.

TASK 5: sampling

As a last preprocessing step, you will try and compare different types of sampling.

5.1

Sample 150 instances (that is, flowers) uniformly at random.

```
sample = data.sample(n=150)
```

5.2

Sample 150 instances using bootstrapping.

```
sample = data.sample(n=150, replace=True)
```

5.3

Compute a stratified sampling of half of the instances, with the sampling probability proportional to the size of the species.

```
sample = data.groupby('species', group_keys=False).apply(
    lambda x: x.sample(frac=0.5))
```

5.4

Compute a stratified sampling of 150 instances, including 50 instances for each species.

```
sample = data.groupby('species', group_keys=False).apply(
    lambda x: x.sample(50))
```