**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code:CSE422** | **Credits:** |
| **Course Name: Artificial Intelligence** | **Prerequisite:** CSE111, CSE221 |

**Lab 03  
Data handling, preprocessing and visualization for AI application**

1. **Lab Overview:**

Data are the facts and statistics collected together for reference or analysis. Data are the inputs for the AI, Mahine learning or Data science methods to find interesting patterns, and/or to learn the insights or to analyze for meaningful information. In this lab session, you will learn:

1. How to load data for processing?
2. How to scale, reshape and normalize data?
3. Statistical feature extraction
4. Encoding and mapping for data quantization
5. Data visualization and plotting
6. **Lesson Fit:**

There is pre-requisite to this lab: CSE111, CSE221. You should have intensive Programming Knowledge and capability to understand algorithms.

1. **Acceptance and Evaluation**

Performed lab tasks will be evaluated by the Lab Instructor (LI)

* 1. Short viva will be conducted in each Lab or occasionally to examine your work.
  2. You may work in groups but be aware that you will be evaluated individually; hence active participation during the Lab work demonstration is recommended.
  3. There will be Lab handout after your work you have to handover it to LI

1. **Learning Outcome:**

After this Lab, the students will be able to:

* 1. Understand the data preprocessing steps
  2. Learn basic feature extraction methods
  3. Perform data plotting and visualization

1. **Activity Detail**
   1. **Hour: 1-2  
      Getting Started:**
      1. Have a glance at Books “Python Data Science Handbook\_ Essential Tools for Working with Data” by Jake VanderPlas, O’Reilly Media (2016)
      2. “Artificial Intelligence with Python written by Prateek Joshi, January 2017
      3. Check \\TSR to see e-book copy, available datasets, codes, and tutorials
      4. https://www.shanelynn.ie/using-pandas-dataframe-creating-editing-viewing-data-in-python/
      5. https://machinelearningmastery.com/prepare-data-machine-learning-python-scikit-learn/

**Used Libraries and APIs:**

**a) Scikit**-**Learn**: There are several Python libraries that provide solid implementations of a range of machine learning algorithms. One of the best known is Scikit-Learn, a package that provides efficient versions of a large number of common algorithms. Scikit-Learn is characterized by a clean, uniform, and streamlined API, as well as by very useful and complete online documentation.

Syntax to import Scikit-Learn:

* + - from sklearn import datasets
    - from sklearn import preprocessing

**b) NumPy**: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Syntax to import NumPy:

* import numpy as np

**c) Pandas (Python** Data Analysis Library): **pandas** is a **Python** package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in **Python.**

Syntax to import **pandas** :

* import pandas as pd

**d)** **Matplotlib**: Thematplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

Syntax to import matplotlib:

* import matplotlib.pyplot as plt

**e) Seaborn:** The seaborn provides an API on top of Matplotlib that offers sane choices for plot style and color defaults, defines simple high-level functions for common statistical plot types, and integrates with the functionality provided by Pandas DataFrames.

Syntax to import seaborn:

* import seaborn as sns

**Discussion: Data representation and data loading**

Machine learning is about creating models from data: for that reason, we’ll start by discussing how data can be represented in order to be understood by the computer. The best way to think about data is in terms of tables of data or matrix. A basic table is a two-dimensional grid of data, in which the rows represent individual elements of the dataset, and the columns represent quantities related to each of these elements.

In general, we will refer to the rows of the matrix as samples, and the number of rows as n\_samples. Likewise, each column of the data refers to a particular quantitative piece of information that describes each sample. In general, we will refer to the columns of the matrix as features, and the number of columns as n\_features. Therefore, the features matrix is assumed to be two-dimensional, with shape [n\_samples, n\_features], and is most often contained in a NumPy array or a Pandas DataFrame, though some ScikitLearn models also accept SciPy sparse matrices.

There are several Python libraries that provide solid implementations of a range of machine learning algorithms. One of the best known is Scikit-Learn, a package that provides efficient versions of a large number of common algorithms. Scikit-Learn is characterized by a clean, uniform, and streamlined API, as well as by very useful and complete online documentation.

**Data loading procedures:**

import pandas as pd

from sklearn import datasets

# loading builtin default from skicit learn

dt = datasets.load\_boston()

print(dt.data)

#import data as dataframe (converting numpy array to dataframe)

df = pd.DataFrame(dt.data, columns=dt.feature\_names)

print(df.head())

#To load local dataset

data = pd.read\_csv('diabetis.csv')

**Data preprocessing and statistical feature extraction procedures:**

import numpy as np

from sklearn import preprocessing

binarized = preprocessing.binarize(input\_data)

# calculation column wise

print("Mean of the given data", input\_data.mean(axis=0))

print("Standard Deviation of the given data", input\_data.std(axis=0))

# After scaling

scaled = preprocessing.scale(input\_data)

print("Mean of the given data", scaled.mean(axis=0))

print("Standard Deviation of the given data", scaled.std(axis=0))

# Minmax scaling

scale\_range = preprocessing.MinMaxScaler(feature\_range=(0,1))

scaled = scale\_range.fit\_transform(input\_data)

print(scaled)

# normalize data

norm\_l1\_data = preprocessing.normalize(input\_data, norm = 'l1')

print("\n normalized L1 data\n",norm\_l1\_data)

norm\_l2\_data = preprocessing.normalize(input\_data, norm = 'l2')

print("\n normalized L2 data\n",norm\_l2\_data)

# Encoding data

encoder = preprocessing.LabelEncoder()

encoded = encoder.fit\_transform(input\_data)

print(encoded)

for i, item in enumerate(encoder.classes\_):

print(item,'-->',i)

# decoding data

decoding = encoder.inverse\_transform(encoded)

print("\n",decoding)

**Data plotting and visualization procedures:**

dt.plot(kind='box', subplots=True, layout=(2,2))

plt.show()

dt.hist()

plt.show()

pd.tools.plotting.scatter\_matrix(dt)

plt.show()

**Activity List**

(It is Not a Group Task, Try Individually)

**Task 01:** Mark 10 **Time:** 1 hour

Refer to the file ‘diabetes.csv’. It shows 8 features and their outcome. The feature values are labeled sequentially as Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, Diabetes Pedigree Function and Age respectively. Results are stored in the Outcome column. All the data are in the numerical format except the Outcome column. It stores YES or NO value which refers to specific patient suffers from diabetes or not while features in each row represents various diagnostic values of a specific patient.

For this dataset, you need to do the following:

1. Read the csv file as a data frame.

2. Encode outcomes as they are not numerical.

3. Divide the frame as data & outcome

4 . Visualize data through plotting

5. Extract basic statistical features

**Evaluation Process (VIVA):** You have to explain your program and show your work to the Lab Instructor. Instructor may ask you some questions.