Lab3

February 7, 2023

Lab 3 has two parts (A) and (B) and you are required to implement it in Python. **A good coding style should be followed, consider PEP 8 [1] or Google Python coding style [2].**

Part A

Naive Bayes, Decision Trees, K-Nearest Neighbors ML Models: Describe and show detailed working of **any one algorithm** from Naive Bayes, Decision Trees and K-Nearest Neighbors (KNN) using corresponding datasets mentioned below. You can use the APIs of your choice. Explain the model outcomes clearly with graphs/plots.

Naive Bayes

Follow the instructions and python code in [3] using scikit-learn on the UCI wine dataset from here [4] and implement in your own virtual environment. For further reading and understanding you can check the Naive Bayes implementation from scratch in Python over here [5].

Decision Trees

Follow the instructions and python code in [6] using scikit-learn on the Diabetes dataset from here [7] and implement in your own virtual environment. For further reading and understanding you can check the Decision Trees implementation from scratch in Python over here [8].

K-Nearest Neighbors

Follow the instructions and python code in [9] using scikit-learn on the UCI wine dataset from here [4] and implement in your own virtual environment. For further reading and understanding you can check the KNN implementation from scratch in Python over here [10].

Part B

Artificial Neural Networks and Clustering: Describe and show detailed working of **any one algorithm** from Artificial Neural Networks and Clustering using the corresponding datasets mentioned below. You can use the APIs of your choice. Explain the model outcomes clearly with graphs/plots.

Artificial Neural Network

Perform the following two tasks:

(i) Follow the instructions and python code in [11] using keras on the Zillow's Home Value Prediction dataset modified here [12] and implement in your own virtual environment. The code is also available here on github [13].

(ii) Follow the instructions and python code in [14] using keras on the same UCI wine dataset from here [15] and implement in your own virtual environment.

Clustering

First follow the instructions and python code in [16] first to understand DBScan and implement the example in python. Then perform the following two tasks.

(i) Follow the instructions and python code in [17] to perform clustering and observe the relationship between Spend Score vs Annual Income while using DBScan. This dataset can be found here [18]. Focus mainly on cell 47 in the notebook and onwards. Try to change the epsilon and min points values and note down your observations.

(ii) Follow the instructions and python code in [19] for only one dataset i.e. the AWS Cloud Watch dataset to perform clustering on time series data to observe anomalies. This dataset is part of Numenta Anomaly Benchmark paper here [20] which can be read for gaining further knowledge about this dataset. Implement this task in your own virtual environment and explain your implementation for multiple ec2 cpu utilization files.

Work Environment

This lab project will be created using a virtual environment in Anaconda to show application level isolation.

Lab Report

The report should be written in the lab report format template file, can be found here [21] (Do not edit this template file, make a copy in your own Overleaf account to edit!) using LaTeX in Overleaf. Download the pdf file after you finish writing and submit the pdf along with your zipped code files to Canvas. **You must document your code properly so its readable.

References

[1] "PEP 8 — Style Guide for Python Code." https://www.python.org/dev/peps/pep-0008/e. Accessed: 2021-02-11.

[2] "Google Python Style Guide." https://google.github.io/styleguide/pyguide.html. Accessed: 2021-02-11.

[3] "Naive Bayes Classification." https://www.datacamp.com/community/tutorials/ naive-bayes-scikit-learn. Accessed: 2021-02-18.

[4] "Wine Dataset for Multinomial Naive Bayes." https://archive.ics.uci.edu/ml/datasets/wine. Accessed: 2021-02-18.

[5] "Naive Bayes from Scratch in Python."https://machinelearningmastery.com/ naive-bayes-classifier-scratch-python/. Accessed: 2021-02-18.

[6] "Decision Trees Classification for Diabetes." https://www.datacamp.com/community/tutorials/ decision-tree-classification-python#comments. Accessed: 2021-02-18.

[7] "Diabetes Dataset for Decision Trees." https://www.kaggle.com/uciml/pima-indians-diabetes-database. Accessed: 2021-02-18.

[8] "Decision Trees from Scratch in Python."https://machinelearningmastery.com/ implement-decision-tree-algorithm-scratch-python/. Accessed: 2021-02-18.

[9] "KNN." https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn, note = Accessed: 2021-02-18.

[10] "KNN from Scratch in Python."https://machinelearningmastery.com/ tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/. Accessed: 2021-02-18.

[11] "ANN for Lot Price Prediction." https://www.freecodecamp.org/news/ how-to-build-your-first-neural-network-to-predict-house-prices-with-keras-f8db83049159/. Accessed: 2021-02-18.

[12] "ANN for Lot Price Prediction." https://drive.google.com/file/d/1GfvKA0qznNVknghV4botnNxyH-KvODOC/ view. Accessed: 2021-02-18.

[13] "Lot Price Prediction Python Code on Github." https://github.com/josephlee94/ intuitive-deep-learning. Accessed: 2021-02-18.

[14] "ANN for Wine Prediction." https://www.datacamp.com/community/tutorials/deep-learning-python. Accessed: 2021-02-18.

[15] "ANN for Wine Quality Dataset." https://archive.ics.uci.edu/ml/datasets/wine+quality. Accessed: 2021-02-18.

[16] "DBScan Basics in Python." https://scikit-learn.org/stable/auto_examples/cluster/plot_dbscan.html. Accessed: 2021-02-18.

[17] "DBScan Spend Score vs Annual Income." https://www.kaggle.com/bagavathypriya/dbscan-clustering. Accessed: 2021-02-18.

[18] "Mall Customer Dataset." https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python. Accessed: 2021-02-18.

[19] "DBScan based anomaly detection for AWSCloudWatch Dataset." https://www.kaggle.com/d4v1d3/dbscan/notebook?select=README.md, note = Accessed: 2021-02-18.

[20] "Evaluating Real-time Anomaly Detection Algorithms – the Numenta Anomaly Benchmark." https://arxiv.org/pdf/1510.03336.pdf. Accessed: 2021-02-18.