Spring 2022: CSEE5590/490 – Special Topics

Python and Deep Learning - ICP-5

Lesson Overview:
In this lesson we will introduce classification.
b. Classification algorithm
c. Scikit learn

Use Case Description:
k-nearest neighbor classifier

Programming elements:
Classification

Data Set:

Dataset: Titanic
Dataset description: Link
Dataset: Optical Recognition of Handwritten Digits Data Set
Dataset description: Link

Assignment:

Q1. (Titanic Dataset**)**

1. Find the correlation between 'survived' (target column) and 'sex' column for the Titanic use case in class.
    a. Do you think we should keep this feature?
2. Do at least two visualizations to describe or show correlations. (e.g.: Survived: Class and gender).
3. Implement SVM method using scikit-learn library and report the accuracy.
4. Implement the multiple types of Naïve Bayes methods using scikit-learn library and report the accuracies
5. Implement KNN method using scikit-learn library and report the accuracy
6. Report the `confusion_matrix` for all the algorithms (SVM, Naïve Bayes, KNN)

Q2. (Optical Recognition of Handwritten Digits Data Set) Link

1. Implement [SVM, Naïve Bayes, KNN] methods using scikit-learn library.
2. Use the digits dataset available in the link above.
3. Visualize some of the images and analyze the data (explore the dataset).
4. Use **train_test_split** to create the training and testing parts.
5. Evaluate the model on testing part using score and the `classification_report()` method.
6. Evaluate an image and visualize its own class.
7. Using KNN algorithm which K value can give good results?

Q3. (10 BOUNS points) (Optical Recognition of Handwritten Digits Data Set) New_Link

1. Download the dataset from the new source.
2. Apply the required preprocessing on the dataset.
3. Explain your implemented preprocessing steps.
4. Implement [SVM, Naïve Bayes, KNN] methods using scikit-learn library.

** Follow the IPC rubric guidelines.

Submission Guidelines:

1. Once finished document your code and make sure all parts of the assignments are completed.
2. Push your code to your GitHub repo and update the ReadMe file, add your info, and partner info.
3. Submit the assignment on Canvas.
4. Present your work to TA during class time to prove the execution and complete submission.

After class submission:

1. Once finished document your code and make sure all parts of the assignments are completed.
2. Push your code to your GitHub repo and update the ReadMe file, add your info, and partner info.
3. Submit the assignment on Canvas before the deadline.
4. Record a short video (1~3) minute, proof of execution and complete assignment.
5. Add video link to ReadMe file.

Note: *Cheating, plagiarism, disruptive behavior, and other forms of unacceptable conduct are subject to strong sanctions in accordance with university policy. See detailed description of university policy at the following URL: https://catalog.umkc.edu/special-notices/academic-honesty/*