Fall 2021: 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:** Glass
>**Dataset description:** https://www.kaggle.com/uciml/glass
>he original glass identification dataset from UCI machine learning repository is a classification dataset.
>The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence, if correctly identified. This dataset contains attributes regarding several glass types (multi-class). The name of target Column is **Type.**

Assignment:

1.   (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 Naïve Bayes method using scikit-learn library and report the accuracy.

2.   (Glass Dataset**)**

1. Implement Naïve Bayes method using scikit-learn library.
    a. Use the glass dataset available in Link also provided in your assignment.
    b. Use **train_test_split** to create training and testing part.
2. Evaluate the model on testing part using score and

```
classification_report(y_true, y_pred)
```

1. Implement linear SVM method using scikit library
    a. Use the glass dataset available in Link also provided in your assignment.
    b. Use **train_test_split** to create training and testing part.
2. Evaluate the model on testing part using score and

```
classification_report(y_true, y_pred)
```

Do at least two visualizations to describe or show correlations in the Glass Dataset.

Which algorithm you got better accuracy? Can you justify why?

Submission Guidelines:

1. Once finished present your work to TA during class time.
2. Once evaluated submit your source code and documentation to GitHub and represent the work in a ReadMe file properly (short summary for the ICP).

After class submission:
1. Complete your work and submit to your repo before the deadline.
2. Record a short video (1~3) minute, explaining the technical part and method used.
3. 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/*