|  |  |
| --- | --- |
| [Machine Learning]  [2021-1] |  |
| Homework 2  Lec 5, 6, 7 |  |
| [Due Date] 2021.04.23  Student ID :  Name :  Professor : Juntae Kim | logo-placeholder |

1. Answer following questions (30 pts)

* 1. Describe the odds ratio, logit, logistic function(including math formula) and their meaning.

odds ratio : 실패확률에 대한 성공확률의 비율

logit : log odds

logistic function : 라 할 때, 에 대해서 수식을 전개하면

* 1. Describe the two Impurity measures for Decision Tree Learning(including math formula) and their meaning.

Entropy: 불확실성의 측정

Gini Index : 랜덤하게 선택된 요소들이 무작위로 라벨링되어 있다면 얼마나 많이 잘못 라벨링되어있는지를 측정하는 지수

* 1. Describe the Naïve Bayesian classifier(including math formula) and its meaning. Also explain how you can deal with the continuous feature values.

|  |
| --- |
| Your Answer |
| Naïve Bayesian Classifer는 모든 특성들이 서로 독립이라는 가정을 하고 Bayes’ Rule에 입각한 probabilistic classifier이다. 즉, 이 주어졌을때 특정 클래스 에 속할 확률은 다음과 같다 **…** 따라서 예측은 다음식을 통해 이루어진다. **(.** Naïve Bayesian Classifier가 연속적인 값들에 대해 적용될 때, 이 각 클래스에 관련된 속성값들이 가우시안 분포를 따른다고 가정한다. 따라서 이 때, 클래스가 주어졌을 때 특정속성값의 확률을 다음과 같다. **. 이것으로 다시 (.**을 이용하여 예측을 수행한다 |

2. Apply Logistic Regression, Decision Tree, Naïve Bayesian Classifier, k-Nearest Neighbor on Wine Dataset to predict the origin of wines. Describe the learned model, and compare the accuracies. (30 pts)

* Dataset

<https://archive.ics.uci.edu/ml/datasets/Wine>

The dataset is the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the 3 types of wines.

* Use downloaded raw data or scikit-learn library

<https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html>

from sklearn.datasets import load\_wine

wine = load\_wine()

X = wine.data

y = wine.target

* Check test accuracy (use 30% for test)

lr.score(X\_test\_std, y\_test)



|  |
| --- |
| Code |
| 첨부한 hw2-2 참고 |
| Result(Captured images) |
|  |
| Description |
| 데이터를 받아오고 train\_test\_split으로 train\_set과 test\_set으로 나눠준다.  DecisionTreeClassifier, GaussianNB, LogisticRegression 객체를 생성한다.  sklearn.preprocessing의 StandardScaler를 이용해서 Standardization해준다.  이제 각 분류모델 객체를 train\_set과 fit메소드로 훈련하고 test\_set과 score메소드로 성능을 측정한다 |

3. Describe why the K-Nearest Neighbors method is not appropriate for dataset with large number of features. (10 pts)

|  |
| --- |
| Your Answer |
| 차원이 늘어날수록 데이터 사이의 거리가 멀어지고, 빈공간이 증가하는 Sparsity를 보인다. 즉, 동일한 개수의 데이터의 밀도가 희박해진다. KNN알고리즘은 유클라디안 거리를 사용하기 때문에 차원이 증가할수록 주어진 관측치에 가까운 이웃이 없는 현상이 발생한다. 따라서 feature수가 엄청 많을 때(차원의 수가 굉장히 클때) KNN알고리즘은 적합하지 않다. |

4. Apply Multi-layer Perceptron on Olivetti Faces Dataset to identify persons from images. Describe the learned model. (30 pts)

* Dataset

<https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_olivetti_faces.html>

The Olivetti Faces dataset contains a set of face images taken at AT&T Laboratories Cambridge. The sklearn.datasets.fetch\_olivetti\_faces function is the data fetching function that downloads the data.

There are 10 different images of each of 40 distinct persons. For some persons, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses).

The 64x64 pixels image is quantized to 256 grey levels and stored as unsigned 8-bit integers; the loader will convert these to floating point values on the interval [0, 1]. The target for this database is an integer from 0 to 39 indicating the identity of the person pictured.

* Check test accuracy (use 20% for test)

print("Training set score: %f" % mlp.score(X\_train, y\_train))

print("Test set score: %f" % mlp.score(X\_test, y\_test))



* Plotting several images (person 0, 1, 2)

from sklearn import datasets

from matplotlib import pyplot as plt

face = datasets.fetch\_olivetti\_faces()

X = face.data

y = face.target

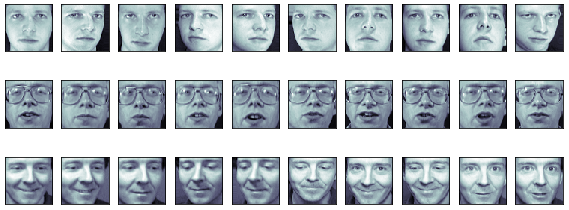
%matplotlib inline

fig = plt.figure(figsize=(10, 4))

for i in range(30):

ax = fig.add\_subplot(3, 10, i + 1, xticks=[], yticks=[])

ax.imshow(face.images[i], cmap=plt.cm.bone)



|  |
| --- |
| Code |
|  |
| Result(Captured images) |
|  |
| Description |
|  |

**Note**

1. Summit the file to e-class as pdf.

2. Specify your file name as “hw2\_<StudentID>\_<Name>.pdf”