Intro. to Machine Learning Project3 Comparison b/w KDTree , Decision Tree and Naive Bayes Classifier Report

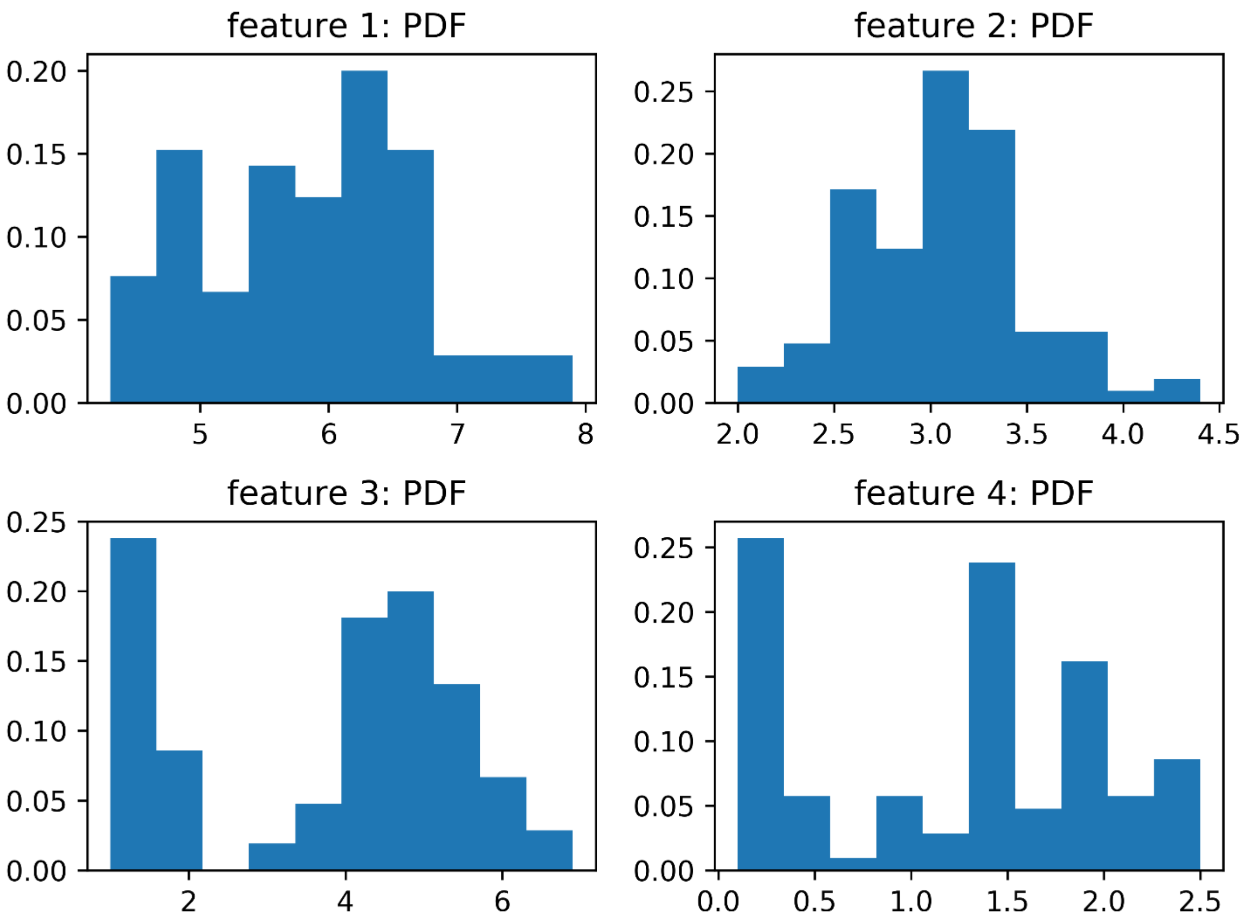
0416324 An-Fong Hwu

Build environment (Note, this report is written in md-like format)
*

```
Python 3.6.3 |Anaconda, Inc.| (default, Oct 13 2017, 12:02:49)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

* Packages including sklearn, numpy, scipy
Where sklearn is used for the constructing/training /validating model, numpy and scipy for the numerical and statistical analysis.

Iris data set



* Probability Distribution Plot of the iris_dataset

* The required Laplacian Smoothing can be found in the Multinomial Naive Bayes Classifier:

1.9.2. Multinomial Naive Bayes

`MultinomialNB` implements the naive Bayes algorithm for multinomially distrit
ıaive Bayes variants used in text classification (where the data are typically rep
f-idf vectors are also known to work well in practice). The distribution is parame
each class y, where n is the number of features (in text classification, the size
$P(x_i \mid y)$ of feature i appearing in a sample belonging to class y.

The parameters θ_y is estimated by a smoothed version of maximum likelihood,

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

where $N_{yi} = \sum_{x \in T} x_i$ is the number of times feature i appears in a sample c
$N_y = \sum_{i=1}^{|T|} N_{yi}$ is the total count of all features for class y.

The smoothing priors $\alpha \geq 0$ accounts for features not present in the learning s
urther computations. Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$

```python
predicted_class_set = []

gnb = MultinomialNB(alpha = 1) #using the "alpha = 1" param setting to set the laplacian smoothing
my_naive_bayes_classifier = gnb.fit(training_set, training_set_predicted)
correct_prediction = 0
predicted_class_set = gnb.predict(testing_set)
```

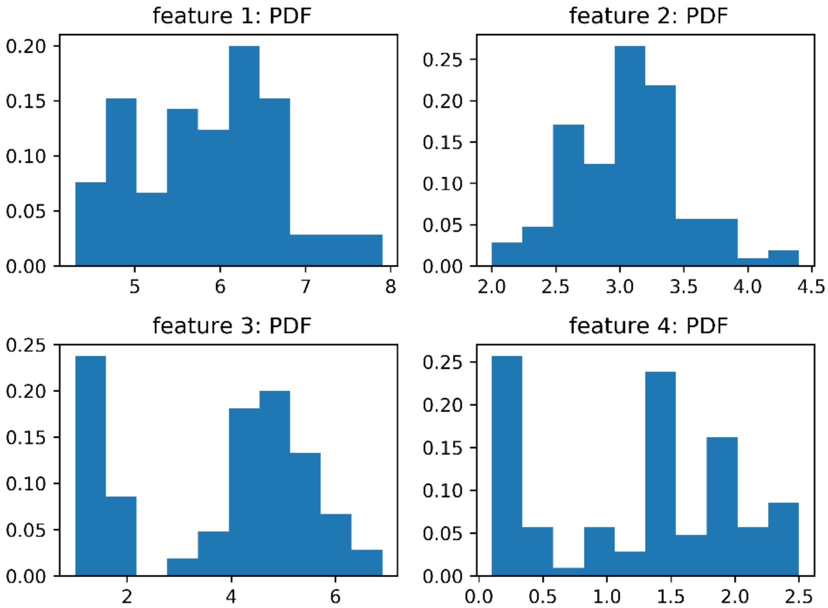
```
Run the model 10 times

1 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 1.0
Multinomial Naive Bayes Classifier Accuracy: 0.9777777777777777
Bernoulli Naive Bayes Classifier Accuracy: 0.28888888888888886
Decision Tree Classifier Accuracy: 0.9555555555555556
KD Tree Classifier Accuracy: 0.9777777777777777

2 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.9555555555555556
Multinomial Naive Bayes Classifier Accuracy: 0.9777777777777777
Bernoulli Naive Bayes Classifier Accuracy: 0.3111111111111111
Decision Tree Classifier Accuracy: 0.9777777777777777
KD Tree Classifier Accuracy: 0.9777777777777777

3 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.9111111111111111
Multinomial Naive Bayes Classifier Accuracy: 0.9111111111111111
Bernoulli Naive Bayes Classifier Accuracy: 0.28888888888888886
Decision Tree Classifier Accuracy: 0.9333333333333333
KD Tree Classifier Accuracy: 0.9333333333333333

4 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.9777777777777777
Multinomial Naive Bayes Classifier Accuracy: 0.6888888888888889
Bernoulli Naive Bayes Classifier Accuracy: 0.26666666666666666
Decision Tree Classifier Accuracy: 0.9555555555555556
KD Tree Classifier Accuracy: 0.9777777777777777

5 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 1.0
Multinomial Naive Bayes Classifier Accuracy: 0.8666666666666667
Bernoulli Naive Bayes Classifier Accuracy: 0.26666666666666666
Decision Tree Classifier Accuracy: 0.9555555555555556
KD Tree Classifier Accuracy: 0.9777777777777777

6 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.9555555555555556
Multinomial Naive Bayes Classifier Accuracy: 0.8666666666666667
Bernoulli Naive Bayes Classifier Accuracy: 0.3111111111111111
Decision Tree Classifier Accuracy: 0.9555555555555556
```

```
7 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.9555555555555556
Multinomial Naive Bayes Classifier Accuracy: 0.9777777777777777
Bernoulli Naive Bayes Classifier Accuracy: 0.26666666666666666
Decision Tree Classifier Accuracy: 0.9333333333333333
KD Tree Classifier Accuracy: 0.9777777777777777

8 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.9777777777777777
Multinomial Naive Bayes Classifier Accuracy: 0.9555555555555556
Bernoulli Naive Bayes Classifier Accuracy: 0.28888888888888886
Decision Tree Classifier Accuracy: 0.9333333333333333
KD Tree Classifier Accuracy: 0.9333333333333333

9 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 1.0
Multinomial Naive Bayes Classifier Accuracy: 0.5777777777777777
Bernoulli Naive Bayes Classifier Accuracy: 0.28888888888888886
Decision Tree Classifier Accuracy: 0.9555555555555556
KD Tree Classifier Accuracy: 0.9555555555555556

10 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.9111111111111111
Multinomial Naive Bayes Classifier Accuracy: 0.8
Bernoulli Naive Bayes Classifier Accuracy: 0.3111111111111111
Decision Tree Classifier Accuracy: 0.9111111111111111
KD Tree Classifier Accuracy: 0.9333333333333333
alfons@alfons  ~/Desktop/Programming/Machine Learning Fall 2017
```

* Observation and my inference from the prediction result of the iris dataset

As we can see, the GaussianNB , KD Tree and Decision Tree Classifier have an outstanding result compared to the Multinominal and Bernoulli NB which they both have a quite inaccurate one. Hence, what is the reason?

●First, consider the probability distribution function of the iris data set



All of the features are rather bearing the resemblance to that of the Normal Distribution, or say the Gaussian Distribution. Namely, the GaussianNB will be quite suitable for the prediction.

But how come the Multinominal and Bernoulli NB produce such an dissatisfying result?

The Multinomial Naïve Bayes model counts how often a certain event occurs in the dataset (for example how often a certain word occurs in a document).

The Bernoulli Naïve Bayes model is similar to the Multinomial Naive Bayes model, but instead of counting how often an event occurred, it only describes whether or not an event occurred (for example whether or not a certain word occurs in a document, where it doesn't matter if it occurs once or 100000 times)

In short, the GNB group the similar data together according to the Gaussian Distribution like mean mean+-std mean+-2std and mean +-3std.

In the other two Naïve Based models, they count each distinct value, even though this is the continuous one, 0.1 0.2 0.3 0.4 will be counted to different type respectively, where they originally should produce the same result. Therefore, it undoubtedly produces a result which is quite inaccurate.

●Then we consider the KD Tree model and Decision Tree model for the iris_dataset
The KD Tree KNN algorithm produces a slightly better accuracy then the Decision Tree while they both produces the results which are quite satisfying.

Def supervised learning (from Wikipedia): Supervised learning is the machine learning task of inferring a function from labeled training data.[1] The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.

And they are both supervised learning since in Decision Tree, it has the classification as input while in KNN, it self-search the NN to find the classification.

●Difference b/w regression and classcification?

Regression involves estimating or predicting a response.
Classification is identifying group membership.
Given the following
$f:x \rightarrow y$
$f:x \rightarrow y$
If y is discrete/categorical variable, then this is classification problem.
If y is real number/continuous, then this is a regression problem.

Iris data set
* Probability Distribution Plot of the forestfire_dataset

```
Gaussian Naive Bayes Classifier Accuracy: 0.6089743589743589
Multinomial Naive Bayes Classifier Accuracy: 0.391025641025641
Bernoulli Naive Bayes Classifier Accuracy: 0.8076923076923077
Decision Tree Classifier Accuracy: 0.7692307692307693
Decision Tree Regressor L2-Norm Error: 730897.01045
Decision Tree Regressor Accuracy(Converted to class using log10 base conversion): 0.7371794871794872
KNN Classifier Accuracy: 0.7307692307692307
NN Regressor L2-Norm Error: 298713.018825
NN Regressor Accuracy(Converted to class using log10 base conversion): 0.6858974358974359

2 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.1794871794871795
Multinomial Naive Bayes Classifier Accuracy: 0.5
Bernoulli Naive Bayes Classifier Accuracy: 0.7564102564102564
Decision Tree Classifier Accuracy: 0.6153846153846154
Decision Tree Regressor L2-Norm Error: 780157.9645
Decision Tree Regressor Accuracy(Converted to class using log10 base conversion): 0.6794871794871795
KNN Classifier Accuracy: 0.6666666666666666
NN Regressor L2-Norm Error: 300264.250425
NN Regressor Accuracy(Converted to class using log10 base conversion): 0.6153846153846154

3 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.4230769230769231
Multinomial Naive Bayes Classifier Accuracy: 0.44871794871794873
Bernoulli Naive Bayes Classifier Accuracy: 0.8141025641025641
Decision Tree Classifier Accuracy: 0.6410256410256411
Decision Tree Regressor L2-Norm Error: 103040.669428
Decision Tree Regressor Accuracy(Converted to class using log10 base conversion): 0.6410256410256411
KNN Classifier Accuracy: 0.7115384615384616
NN Regressor L2-Norm Error: 535867.32635
NN Regressor Accuracy(Converted to class using log10 base conversion): 0.6730769230769231

4 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.21153846153846154
Multinomial Naive Bayes Classifier Accuracy: 0.48717948717948717
Bernoulli Naive Bayes Classifier Accuracy: 0.8012820512820513
Decision Tree Classifier Accuracy: 0.6474358974358975
Decision Tree Regressor L2-Norm Error: 1990503.26413
Decision Tree Regressor Accuracy(Converted to class using log10 base conversion): 0.6730769230769231
KNN Classifier Accuracy: 0.6858974358974359
NN Regressor L2-Norm Error: 1481437.3182
NN Regressor Accuracy(Converted to class using log10 base conversion): 0.6410256410256411
```

Up without round (only int) after log Down with round after log, round to the closest integer

```
Gaussian Naive Bayes Classifier Accuracy: 0.23717948717948717
Multinomial Naive Bayes Classifier Accuracy: 0.40384615384615385
Bernoulli Naive Bayes Classifier Accuracy: 0.5769230769230769
Decision Tree Classifier Accuracy: 0.4935897435897436
Decision Tree Regressor L2-Norm Error: 2066317.03684
Decision Tree Regressor Accuracy(Converted to class using log10 base conversion): 0.47435897435897434
KNN Classifier Accuracy: 0.48717948717948717
NN Regressor L2-Norm Error: 1322604.622
NN Regressor Accuracy(Converted to class using log10 base conversion): 0.46794871794871795

2 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.1858974358974359
Multinomial Naive Bayes Classifier Accuracy: 0.391025641025641
Bernoulli Naive Bayes Classifier Accuracy: 0.6538461538461539
Decision Tree Classifier Accuracy: 0.4807692307692308
Decision Tree Regressor L2-Norm Error: 340160.391225
Decision Tree Regressor Accuracy(Converted to class using log10 base conversion): 0.4807692307692308
KNN Classifier Accuracy: 0.4935897435897436
NN Regressor L2-Norm Error: 454092.73675
NN Regressor Accuracy(Converted to class using log10 base conversion): 0.42948717948717946

3 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.20512820512820512
Multinomial Naive Bayes Classifier Accuracy: 0.33974358974358976
Bernoulli Naive Bayes Classifier Accuracy: 0.6538461538461539
Decision Tree Classifier Accuracy: 0.5576923076923077
Decision Tree Regressor L2-Norm Error: 641547.977969
Decision Tree Regressor Accuracy(Converted to class using log10 base conversion): 0.5448717948717948
KNN Classifier Accuracy: 0.5576923076923077
NN Regressor L2-Norm Error: 948852.821875
NN Regressor Accuracy(Converted to class using log10 base conversion): 0.48717948717948717

4 th time training and validation
Gaussian Naive Bayes Classifier Accuracy: 0.1987179487179487
Multinomial Naive Bayes Classifier Accuracy: 0.40384615384615385
Bernoulli Naive Bayes Classifier Accuracy: 0.5897435897435898
Decision Tree Classifier Accuracy: 0.4807692307692308
Decision Tree Regressor L2-Norm Error: 251232.869725
Decision Tree Regressor Accuracy(Converted to class using log10 base conversion): 0.4807692307692308
KNN Classifier Accuracy: 0.5192307692307693
NN Regressor L2-Norm Error: 473134.689675
NN Regressor Accuracy(Converted to class using log10 base conversion): 0.4166666666666667
```

* The required Laplacian Smoothing

```
predicted_class_set = []

gnb = MultinomialNB(alpha = 1) #using the "alpha = 1" param setting to set the laplacian smoothing
my_naive_bayes_classifier = gnb.fit(training_set, training_set_predicted)
correct_prediction = 0
predicted_class_set = gnb.predict(testing_set)
```

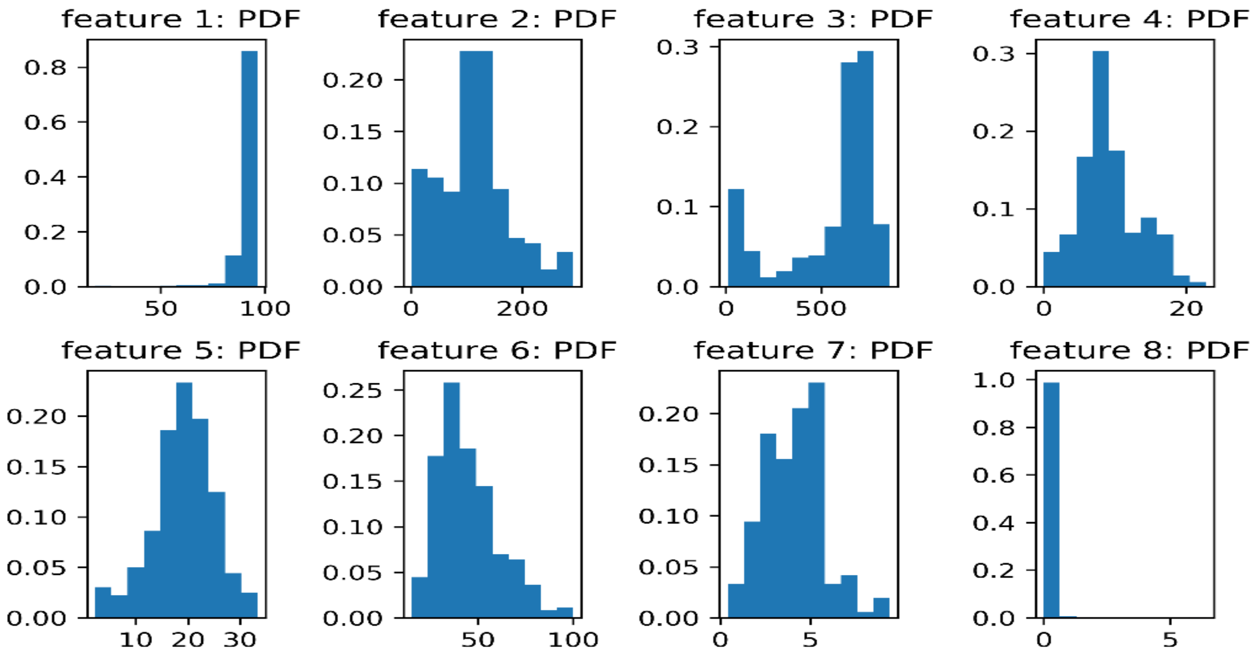
* Observation and my inference from the prediction result of the iris dataset

● First we can see the result with only converting from float to int generates the better result than the training set with round to the nearest int.

Never the less, they still produce the similar AE, the absolute in the L2 norm space of distance.

Reason is that the log + int will cut out all the float part after the integer while log + round will make such logarithmic interval to become more strictly-classified.
e.g. log 90 = 1.95, w/o wound it becomes 1, but w round it will be converted to 2, the classification process will consequently drop.
However, the absolute error does not count in the discrete level, it counts as the continuous level, thus both of the model will produce the similar AE.

●



As we can see, the PDF of forestfire dataset is quite unevenly distributed.
Hence for the GaissianNB which takes the presumption that the dataset is in Gaussian distribution (only feature 1 4 5 6 are close to Gaussian distribution) failed to be accurate.

Due to the unevenly distributed dataset, the KDTree, Decision Tree Classifier ,NN Regressor and DT Regressor all generate the inaccurate result.

BernoulliNB surprisingly result in an outstanding accuracy, the reason I guess is that dataset in forestfire although unevenly distributed, but Bernoulli only check (not count) the occurrence or not(namely happened or not, binary-like processing) in the dataset.

MultinomialNB counts the times of occurrence in the multinomial distribution, compared to the BernoulliNB, it maybe classify the dataset too strictly (sometimes the strict classification can prevent the overfitting problem while this time we did not see such result.)

Overall, the difference b/w BernoulliNB and MultinomialNB mainly lies in “how they count the dataset.” The former only counts whether the event happens or not while the latter counts the OCCURANCE of such event.
Conclusion
In short, suppose we only care whether an event happened or not, we use BerboulliNB, taking a step further we may use the times of occurrence of an event using for analyzing, classification we use MultinomialNB, last but not least, for the dataset which resembles normal distribution, we use GaussianNB.
For the discrete prediction, using the classifier, and using regressor for continuous prediction. (Continuous dataset may use some mapping method to transform into discrete one such as log and round)