Data Warehousing And Data Mining

Major Project - Phase 4

INSTRUCTOR: Dr. Kamalakar Karlapalem

- 20161086, 20161159

Dataset Description

We have chosen the iris dataset for this phase.

Attributes and Class labels

Each sample contains 4 features - *sepal length*, *sepal width*, *petal length*, *petal width* of a type of an Iris plant. Each sample is labelled as - **Iris-setosa**, **Iris-versicolor**, **Iris-virginica**. There are 150 records in the dataset. All the lengths and widths are in cms.

Data Visualization

Data Visualization code available here.

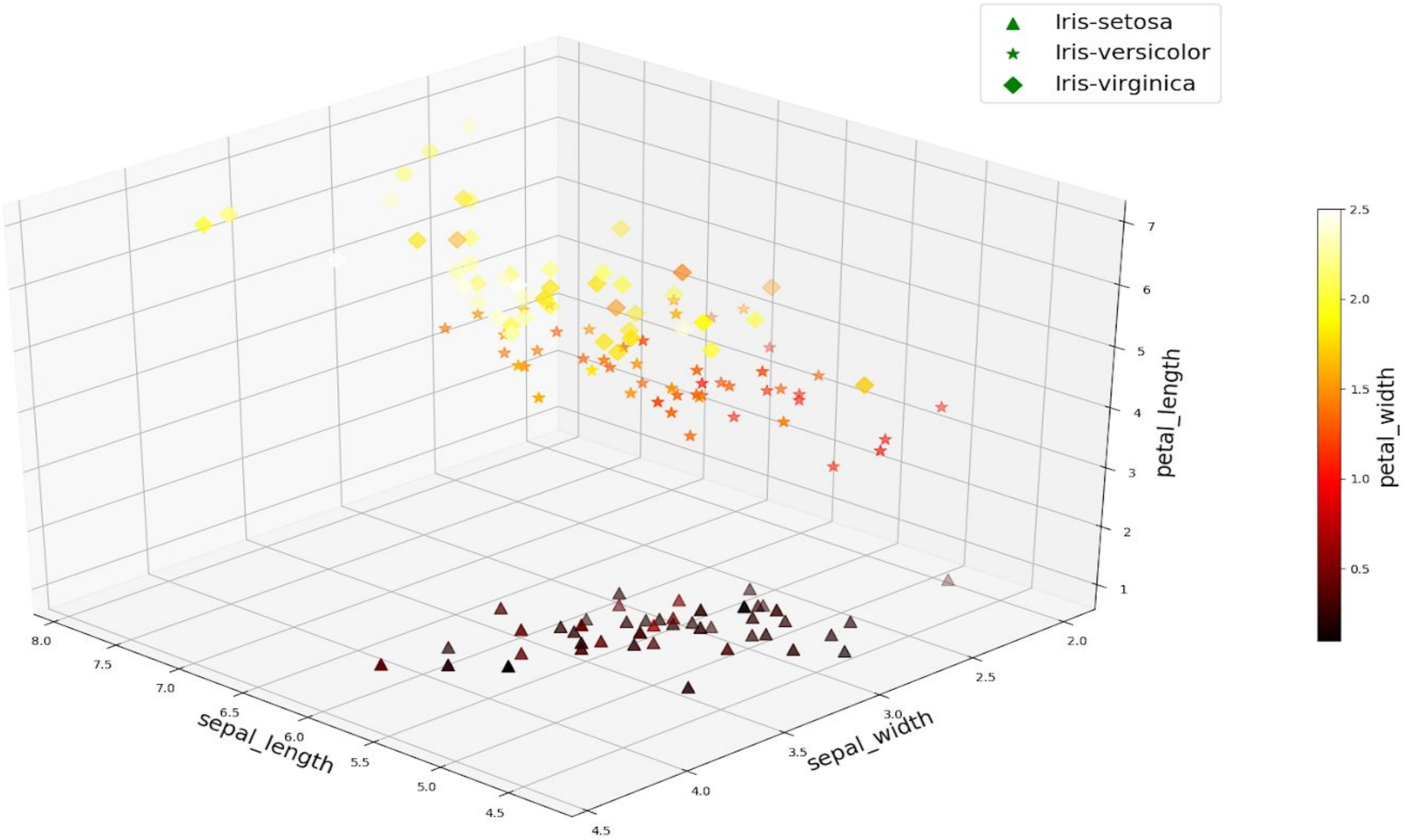

Fig. 1 (Dataset visualization) *Click image to view in browser*

Classification Algorithm used

Naive Bayes Classifier:

We used Naive Bayes Classifier to classify each iris plant. It is based on the bayes theorem and assigns a sample to the class with maximum likelihood.

Why Naive Bayes?

The features - Sepal width, Sepal height, Petal width and Petal height are independent of each other and also contribute equally in determining the type of iris plant, hence we've used the naive bayes classifier. Also since the zero conditional probability case will not arise, we didn't have to do any corrections as well, thus making our classification - fast, accurate and independent of any custom parameters. Due to the fast training and testing time, we were able to estimate it's accuracy exhaustively.

Implementation

We implemented the algorithm using python3, completely from scratch (without the use of any external tool). The code is available as ***holdout.py*** and ***cross-validation.py*** in this folder.

Classifier Evaluation Metrics

We used different methods and techniques to estimate our classifiers' accuracies. Following were the methods and the results:

Holdout Method

We partitioned our dataset into training data containing 70% of the samples and test data containing 30% of the samples. We did this random sampling 1000 times and got the following results:

Confusion Matrix

After 1000 iterations of randomly splitting the dataset into 70% training and 30% testing, we got the below confusion matrix. Since the number of entries in the dataset is 150, total number of test samples over all iterations is 0.3*150*1000 = 45000.

Actual class/ Predicted class	**Iris-setosa**	**Iris-versicolor**	**Iris-virginica**	**Total = 45000**
Iris-setosa	14965	0	0	**14965**
Iris-versicolor	0	13908	1071	**14979**

Iris-virginica	0	1118	13938	**15056**

Precision

Iris-setosa

Precision = 14965/(14965+0+0) = 1

Iris-versicolor

Precision = 13908/(0+13908+1118) = 0.925

Iris-virginica

Precision = 13938/(0+13938+1071) = 0.925

Recall

Iris-setosa

Recall = 14965/(14965+0+0) = 1

Iris-versicolor

Recall = 13908/(0+13908+1071) = 0.928

Iris-virginica

Recall = 13938/(0+13938+1118) = 0.9257

F-score

Iris-setosa

F-score = 2/2 = 1

Iris-versicolor

F-score = 1.7168/1.853 = 0.9264

Iris-virginica

F-score = 1.712/1.85 = 0.9254

Accuracy

Minimum Accuracy	84.44%
Maximum Accuracy	100.0%
Average Accuracy	95.3%

Cross Validation Method

Since our data set was small, we used the leave out 1 method to cross validate. We left each sample out once, and trained our classifier on the remaining models. Using this method, 143 out of 150 samples were classified correctly and hence the accuracy was 95.33%.

Confusion Matrix

After 150 iterations of leave-out-1 testing (each sample is left out and tested against the model trained by the remaining 149 samples), we got the underneath confusion matrix.

Actual class/ Predicted class	**Iris-setosa**	**Iris-versicolor**	**Iris-virginica**	**Total = 150**
Iris-setosa	50	0	0	**50**
Iris-versicolor	0	47	3	**50**
Iris-virginica	0	4	46	**50**

Precision

Iris-setosa

Precision = 50/(50+0+0) = 1

Iris-versicolor

Precision = 47/51 = 0.921

Iris-virginica

Precision = 46/49 = 0.938

Recall

Iris-setosa

Recall = 50/(50+0+0) = 1

Iris-versicolor

Recall = 47/50 = 0.94

Iris-virginica

Recall = 46/50 = 0.92

F-score

Iris-setosa

F-score = 2/2 = 1

Iris-versicolor

F-score = 1.731/1.861 = 0.93

Iris-virginica

F-score = 1.726/1.858 = 0.9289

Accuracy

Accuracy of the classifier using the cross validation method was 95.3%.