

**TDS3301 – DATA MINING**

**Lecturer: Dr. Ho Chiung Ching**

**Tutorial Section: TT01**

**Assignment #3: Classification**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| 1132701439 | Tay Tiong Soon |
| 1132700475 | Teo Yong Zhi |
| 1131122661 | Tazeek Bin Abdur Rakib |
| 1132701076 | Khairul Azmi bin Aminuddin |

**Exploratory Data Analysis**

The Otto dataset was used for our assignment. The dataset consists of more than 60,000 products, with **93** features per product. There are **9 different** types of products, hence, this can be considered a multiclass classification problem. Figure 1 shows the distribution of the classes in the dataset.

To find similarity between products, we reduce the dimensionality size from to only two-dimensions by using by constructing a probability distribution over pairs of words such that similar words with high probabilities will be closer together. This is shown in Figure 2.

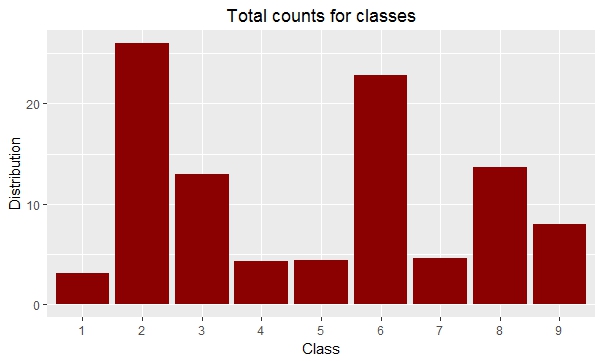


Figure 1: Class distribution

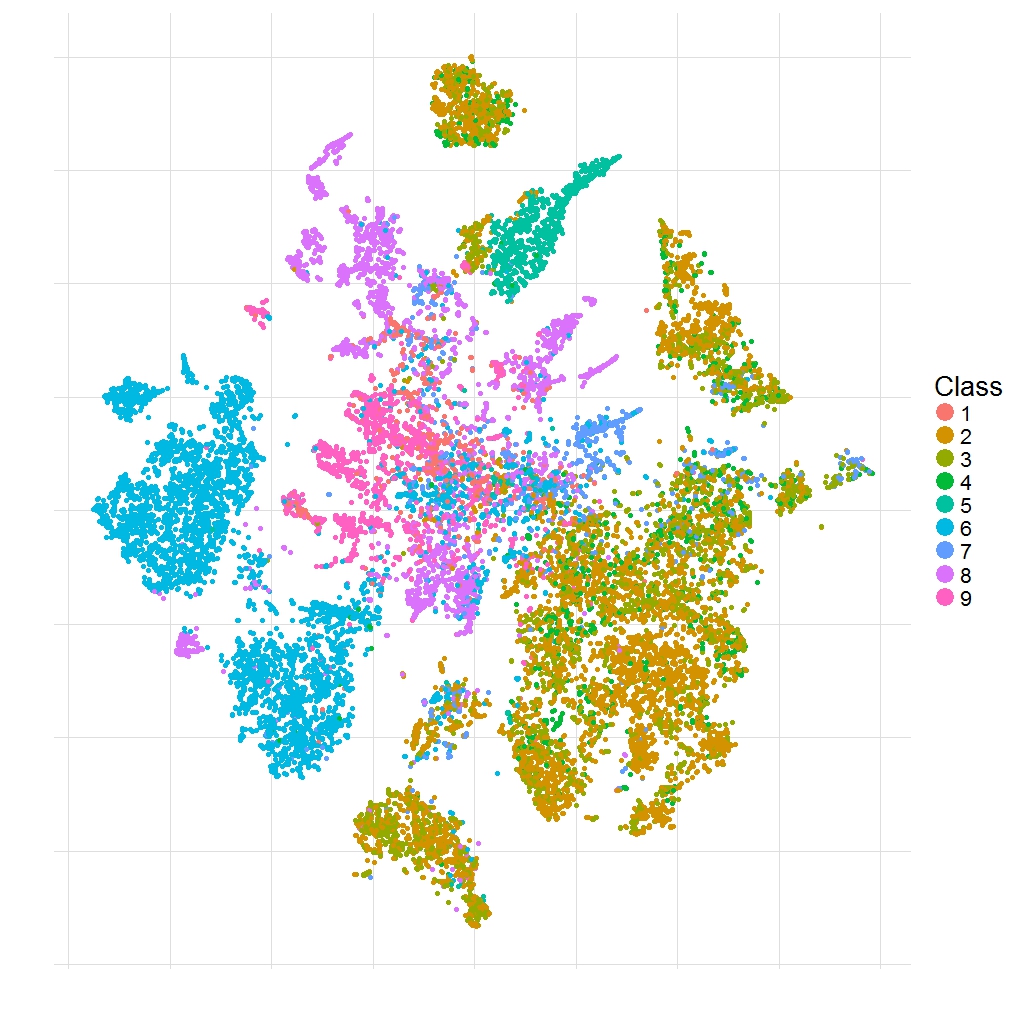


Figure 2: Visualization of products using t-SNE

**Preprocessing Tasks**

The following pre-processing tasks were performed:

* Normalization: Rename the class labels from, for example, ‘Class\_1’ to 1
* Drop the ‘id’ column since it provides no contribution
* For Neural Network only: the data was scaled for training the architecture.

**Evaluation Metrics**

Since it is a multiclass classification, we would take the following metrics into consideration for each class:

* **Precision:** The capability of a classifier not to label a sample as positive when
* its actual value is negative
* **Recall:** Recall is a measure of how many of the correct hits were found.
* **F-1 Measure:** It is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score.

**Results & Discussion**

The dataset is partitioned into 70% training and 30% testing. We plot the metrics’ scores for each classifier, as well as the resulting heatmap to visualize the distribution labels that were classified correctly and wrongly.

*a) Naïve Bayes*

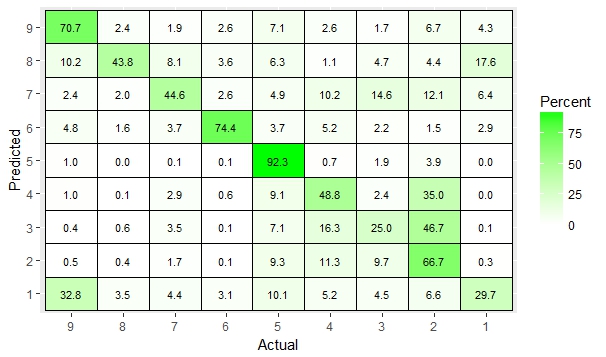


Figure 3: Confusion Matrix of Naïve Bayes

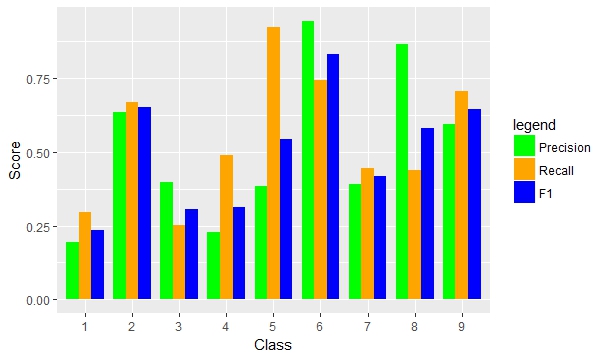


Figure 4: Naïve Bayes Classification results

*b) Artificial Neural Network (ANN)*

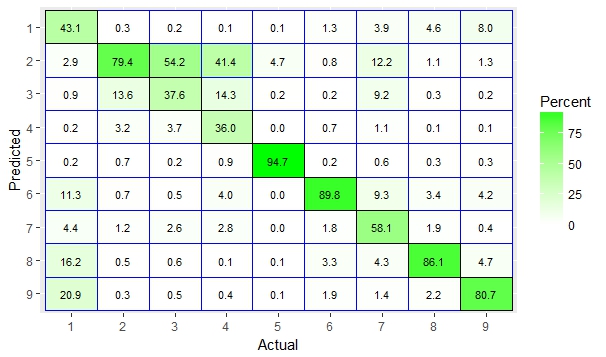
**

Figure 5: Confusion Matrix of ANN

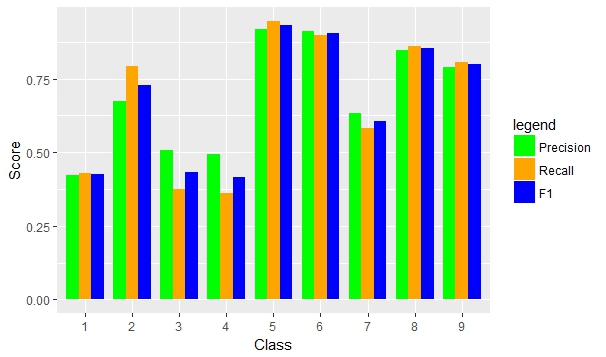
**

Figure 6: ANN Classification Results

*c) Random Forest*

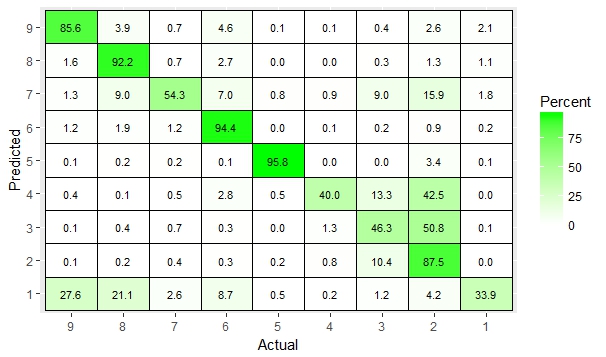


Figure 7: Confusion Matrix of Random Forest

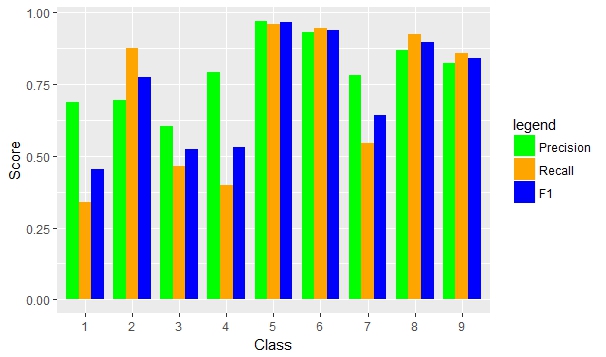


Figure 8: Random Forest Classification Results

From the generated results, we can see that:

* In Naïve Bayes, five out of the nine products were not classified properly, as their classification scores were less than 50%, as shown in Figure 2. The products and their classification score were: 1 (29.7%), 3 (25%), 7(44.8%), and 8(43.8%).
* From Figure 3, it can be seen that products with target 1, 3, 4, and 7 gave poor scores, as none the three metrics crossed 50%. Product 6 has the best scores among the 9 classes, as all three metrics are at least 75%.
* When comparing Figures 2 and 4, it can be seen that there is an overall increase in performance for classification. This trend can also been observed in Figure 5, where the results have improved significantly in comparison Naïve Bayes model.
* When random forest is implemented, products 2, 5, 6, 8, and 9 obtain the best possible classification scores, as shown in Figure 7. This can also be justified from Figure 8, where the average metric score is at least 70%.

The classifiers behave differently because:

* From Figure 2, it can be observed that there are a mixture of a multiple classes in various regions. This introduces some noise and affects the classifiers, especially Naïve Bayes since it relies most on discrete counts.
* In ANN, at least half of our classes are generalized well enough to be properly labeled. This could be down how the weights are updated, hence, being robust enough to properly label the data in comparison the other two methods.
* Random Forest is also robust enough to handle the different varieties of products based on features. Since we used 25 trees, each tree gets a subsample of the given training set, as well as a random number of features. This allows it to achieve a slightly better performance than ANN.