Lacey Perry & Gordon Ngo

Project Report

**Abstract**

Our team’s goal is to explore and provide analysis of the performance and effectiveness of supervised learning algorithms - particularly classification algorithms. We are interested in imbalanced data specifically. As a result of this research problem, we seek to gain insight on which techniques/approaches/algorithms are best utilized on various types of datasets and problems, and then develop a report of our findings. Resulting analysis will test the classification algorithms for examination - for example: an imbalanced dataset and evaluating/determining the best approach based on performance.

**Our Methodology**

We tried to solve the problem of the Bayes classifier by breaking it down into smaller units. First, we collected data from the .csv files with the csv reader. Then we used the stored data to count the number of attributes that occurred and under what class. They were incremented by ClassAttributePair objects to hold them together easily. Continuous data was averaged and split on the mean so that it could be classified. This data was all passed into another class to calculate the probabilities of each attribute occurring with each class. All the probabilities were calculated but smoothing was not completely implemented. We ended up just smoothing the data when the result was 0. This did not extend to smoothing all the other data as it should have. Once probabilities were calculated we ran the data as a set of folds. Using k-fold cross validation where k=10%. I set up the data so that it could be easily folded out. Each fold had its own table of data that had to be later combined, leaving out only the data from the 10% that was the test set each time. We were rather pleased with the results to this point.

For evaluation, we tried to write our own Confusion Matrix but it did not function entirely correctly so we produced inaccurate results compared to the algorithms actual findings when run and checked. The results here were disappointing.

**Experimental Report**

As specified in our proposal, or team ran 3 native classification algorithms in WEKA for the 10 datasets that we chose. In order to evaluate the performance of the supervised classification algorithms and conduct the experiment we first had to identify the metrics. Below are metrics we will be focusing on across all three algorithms.

* General Performance Measure: 10-fold cross-validation
* Classification Metrics (below)
* Accuracy
* ROC Area (area under the ROC curve)
* F-measures
* Kappa Statistic
* Recall
* Speed
* Confusion Matrix (results)

F-measures and recall were included specifically for evaluation for possible imbalanced datasets. The rest of the metrics or parameters are meant to provide further insight on our analysis of the classification algorithms and for performance comparison.

Experimental Setup:

Using these metrics, we use the same classification parameters and settings across all the datasets for each of the 3 algorithms to maintain consistency. The result buffers are also included in our zip for individual runs providing further detail.

* Native Naive Bayes Classifier in WEKA with default settings
* Native J48 Decision Tree Classifier in WEKA (CLI - J48 -C 0.25 -M 2)
* Native Logistic Regression Classifier - default function in WEKA (CLI -R 1.0E-8 -M 1 -num-dec-plc 4)

Preprocessing of the datasets were also done via excel and WEKA’s file viewers. For some datasets, some attributes were removed which was done to reduce dimensionality. They were then converted to the .arff format to be easily read in.

Dataset Reports and Analysis

**Dataset 1: Used Car Evaluation**

Summary: In this dataset, our classification goal is to predict a used car’s condition (unacceptable, acceptable, good, very good) based on the attributes.

Interpretation: We analyzed the results and concluded that the best classification algorithm was logistic regression based on the metrics. Overall accuracy on all 3 algorithms were relatively high (85-93%). The logistic regression classifier was doing better than random “chance” than the others as it had a higher kappa statistic and ROC area. The recall was high, more evenly distributed and reflective of its precision which turns our attention to its F-measure. Logistic regression had the highest F-measure meaning it was precise and recall was good. We notice this is relevant and important because the dataset was a bit imbalanced as most of the used vehicles were in the lower ranges of quality (unacceptable and acceptable). Having observed that consistent metrics were consistent via 10-fold cross validation, we determined it was the most accurate algorithm for solving the classification task/problem of this dataset.

Winner: Logistic Regression

**Dataset 2: Spine and Vertebral Abnormality**

Summary: In this dataset, our classification goal is to predict whether a patient’s spinal column is normal or abnormal based on numerical attributes of their physical measurements.

Interpretation: We analyzed the results and concluded that the best classification algorithm was logistic regression based on the metrics. Overall, all three algorithms had accuracy in the ranges between 77-84%. The classifier was doing better than random “chance” than the others as it had a higher kappa statistic and ROC area. The recall was high, more evenly distributed and reflective of its precision than the others as well. This again turns our attention to its F-measure which was the highest of the three. There wasn’t much of an imbalance in this dataset, but these metrics still tell us that it was more accurate than the other two algorithms. Because of these consistent metrics in 10-fold cross validation, we determined it was the most accurate algorithm for solving the classification task/problem of this dataset.

Winner: Logistic Regression

**Dataset 3: HR Analytics - Employee Performance**

Summary: In this dataset, our classification goal is to predict an employee’s performance based on the various evaluations they received at IBM, and other characteristic attributes.

Interpretation: We analyzed the results and concluded that the best classification algorithm was the J48 decision tree on the metrics. Overall, all three algorithms had poor accuracy in the ranges between 48-55%. The dataset contained many more instances than the previous datasets above, and also more attributes that were generic in the sense that they may have little to no correlation to performance. We could have used preprocessing strategies like reducing dimensionality to remove these attributes but felt that it may limit insight and discovery. The J48 classifier had a slightly lower kappa statistic but higher ROC area than the other two algorithms, we interpreted this as meaning the decision tree may have been doing slightly better by random chance. This is probably however its deviation from the other two algorithm’s respective kappa statistic was not much, and made it for it with a higher ROC area. In addition, the F-score was still above the other three.The dataset was fairly balanced as well which further implies the performance on the classification algorithms remained low. In conclusion, we determined J48 decision tree was the most accurate algorithm for solving the classification task/problem of this dataset after comparing their resulting metrics reports.

Winner: J48 Decision Tree

**Dataset 4: HR Analytics - Employee Attrition**

Summary: In this dataset, our classification goal is to predict an employee’s attrition at IBM based on the various evaluations they received at IBM, and other characteristic attributes.

Interpretation: We analyzed the results and concluded that the best classification algorithm was Naive-Bayes based on the metrics. Overall, all three algorithms had similar accuracy in the ranges between 78-83%. The J48 classifier had higher accuracy, but no kappa statistic and lower ROC area than the other two algorithms, we interpreted this as meaning the decision tree was doing slightly better by random chance. In addition, the F-score for J48 was lower than the other three. Logistic Regression on the other hand trailed behind Naive-Bayes for all of these metrics.In conclusion, we determined Naive-Bayes was the most accurate algorithm for solving the classification task/problem of this dataset.

Winner: Naive-Bayes

**Dataset 5: Mushroom Identification**

Summary: In this dataset, our classification goal is to predict the identify a mushroom between two species based performance based their physical and characteristic attributes.

Interpretation: We analyzed the results and concluded that the best classification algorithm was the J48 decision tree on the metrics. Overall, all three algorithms had very high accuracy in the ranges between 95-100%. The dataset was friendly and nice to use because it was balanced and had no problems with missing information. Both the J48 Decision Tree classifier and Logistic Regression produced a 100% accuracy, thus all metrics being 1 and a perfect confusion matrix after running 10-fold cross validation. We were intrigued by this and used percentage split, however this is mostly likely due to some type of unique identifier in the dataset or overfitting. An interesting note is how Naive Bayes classifier which did not get 100% is highlights its difference between itself and Decision Trees and Logistic Regression. We decided to keep this result, as Decision Trees and Logistic Regression have models that split feature space differently, have different model assumptions, etc. Naive Bayes has a disadvantage in some sense because it assumes all features are conditionally independent and its 95% accuracy may not stick out as suspect where the other two do that may alert one to the clues that there could be overfitting and other problems. Because of this, and to remain consistent to our metrics of evaluating performance, we will conclude that J48 Decision Trees as the best algorithm (Logistic Regression was slower).

Winner: J48 Decision Tree

**Dataset 6: Student Performance**

Summary: In this dataset, our classification goal is to predict a student’s performance based on the various evaluations, scores, and other characteristic attributes.

Interpretation: We analyzed the results and concluded that the best classification algorithm was Naive-Bayes based on our experiment metrics. Overall, all three algorithms had a disparity accuracy in the ranges between 48-71%. The Naive-Bayes classifier had greater accuracy than the other three while maintaining the highest Kappa statistic, precision, recall, F-measure, ROC area, and was completed classification the fastest. From the dataset, due to the types attributes and being more discretized in the dataset, it may have been a more advantageous or more appropriate for the Naive-Bayes classifier.

Winner: Naive-Bayes

**Dataset 7: Income Levels in SF Bay**

Summary: In this dataset, our classification goal is to predict a person’s income-level based on the various attributes such as age, profession, education, etc.

Interpretation: We analyzed the results and concluded that the best classification algorithm was Naive-Bayes based on our experiment metrics. Overall, all three algorithms had a similar accuracy in the range ~80-83%. The Naive-Bayes classifier had greater accuracy than the other three while maintaining the highest Kappa statistic, precision, recall, F-measure, ROC area, and was completed classification the fastest. Logistic Regression took the most time to produce results and J48 classifier had a much lower kappa statistic as well as a slightly lower F-measure, recall, precision and overall accuracy. From the dataset, due to the types attributes and being more discretized in the dataset, it may have been a more advantageous or more appropriate for the Naive-Bayes classifier.

Winner: Naive-Bayes

**Dataset 8: Falling in elderly patients**

Summary: In this dataset, our classification goal is to predict an elderly person’s likelihood of falling based on the various attributes representing measurements of blood pressure, sugar, etc. while doing 6 different activity types: (0=Standing 1=Walking 2=Sitting 3=Falling 4=Cramps 5=Running)

Interpretation: We analyzed the results and concluded that the best classification algorithm was Naive-Bayes based on our experiment metrics. Overall, all three algorithms had a similar accuracy in the range  28-65%. The Naive-Bayes classifier had a substantially greater accuracy than the other three while again maintaining the highest Kappa statistic, precision, recall, F-measure, ROC area, and was completed classification the fastest. Logistic Regression took an extraordinarily long time to produce results due to the large dataset and the J48 classifier had a much lower kappa statistic as well as a slightly lower F-measure, recall, precision and overall accuracy. From the dataset, due to the types attributes and being more discretized in the dataset, it may have been a more advantageous or more appropriate for the Naive-Bayes classifier again as well.

Winner: Naive-Bayes

**Dataset 9: Backorders**

Summary: In this dataset, our classification goal is to predict a product’s likelihood getting placed in for a backorder, which is a common logistical problem at many retailers and warehouses based on a variety of attributes such as sales.

Interpretation: We analyzed the results and concluded that the best classification algorithm was Naive-Bayes based on our experiment metrics.The primary deciding factor and weight was speed. Both the logistical regression and decision tree algorithms required a substantial amount of time due to the datasets large size and number of attributes. Overall, all three algorithms had a similar accuracy in the high 94+% range, with logistical regression at the highest accuracy. However, each of the 3 algorithms marginal difference in accuracy as well as all other metrics (Kappa statistic, precision, recall, F-measure, ROC area) were very low. Because Naive-Bayes completed classification the fastest we decide to rank it higher in evaluation because of its much higher speed performance

Winner: Naive-Bayes

**Dataset 10: Video-game Sales**

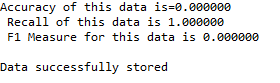
Summary: In this dataset, our classification goal is to predict the genre of a video game based on two attributes: publisher and total sales. Our team had to reduce dimensionality by removing specific geolocation sales attributes as well as the platform attribute because it implies a publisher (Nintendo for Nintendo console lines, Microsoft Studios for such, etc.). We felt though that the publisher was still an important attribute to keep as game development studios are very much related to sales success. The problem here was that there are many, which inflate the number of class labels.

Interpretation: We analyzed the results and concluded that the best classification algorithm was the J48 Decision Tree based on our experiment metrics. Overall, all three algorithms had a low accuracy in the 30-34% range. It also did not help that this was again partially a result of a stubborn dataset with a large size, high dimensionality, too many class labels, etc. Logistic Regression ran into similar problems once again with the dataset being so large, it took an incredible amount of time to run the algorithm even with parameter modification and sampling. We did not include such in our experiment simply because we need to preserve consistency. J48 Decision Tree classification had the highest values for all the metrics (Kappa statistic, precision, recall, F-measure, ROC area) and was completed in a reasonably fast amount of time. It can be seen quite easily when compared to Naive-Bayes, that its confusion matrix was not as terrible.Because of its overall maintenance and lead in metrics, the best algorithm for classification here is the J48 decision tree.

Winner: J48 Decision Tree

Running our own algorithm on the data provided disastrous results due to the confusion matrix not being set up properly and us not getting smoothing implemented. For most datasets accuracy was around 0% while recall was 1.0.

Here is an example run



Overall Analysis Summary

Based on our results, one can view them in the results folder for exact numeric values on the metrics that is included in our submission zip, the ranking of our algorithms are the following for the 10 datasets:

1. Naive-Bayes
2. Decision-Trees / Logistic Regression

Naive-Bayes was the go-to classifier as it was able to applied easily and quickly (speed) on all of the datasets. In terms of the metrics, it had the highest leading margin as well for most of the experiments. J48 Decision Tree and Logistic Regression were ranked as a tie because they came close in terms of accuracy and other metrics like kappa statistic, recall/precision and F-measures, however J48 still surpasses Logistic Regression in the speed department. Overall the experiment helped reinforce our knowledge of these 3 different classifiers as well as understanding when they are best or most effectively used on various types of datasets. Logistic Regression produces great results but can be hefty to run on large datasets with increased dimensionality, so it takes a shot at performance in terms of speed. We learned that J48 decision trees and Naive-Bayes are good go-to and quick classifiers to be used if results are needed quickly. In the end we decided to use WEKA over SK-Learn simply because preprocessing and reading data created some issues, as well as python running different implementations of the above 3 algorithms which J48 decision trees was less efficient in getting a higher accuracy than WEKA.

**Contribution:** Lacey Perry tried to write the algorithms for Naïve Bayes while Gordon Ngo ran the evaluation of the other metrics on the 10 datasets. We split the proposal and presentation work one to one.