Experiments on Diabetes Dataset

Prashanth Rajendran, Rahul Thakkar, Sai Kaushik

Data Pre-Processing:

The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes [1].

- The readmitted attribute values replaced with 1('< 30') and 0('NO' and '>30').
- The ICD-9 codes of the attributes diag1, diag2 and diag3 are converted to their respective categorical attributes given in the research article [2].
- Following attributes are removed from the data set and reasons to do so:
    - Encounter_id : It's an id and doesn't mean anything to readmission.
    - Patient_nbr: It's an id and doesn't mean anything to readmission.
    - Weight: Almost 98% of records has missing value.
    - Payer_code: It's a field related to payment and doesn't mean anything to readmission.
    - examide: All the instances has the value "no"
    - citoglipton: All the instances has the value "no"
- The missing values of categorical/discrete valued attributes are replaced by mode.
- The missing values of continuous valued attributes are replaced by mean.

Evaluation Criteria:

In the given data set the instances with class 0 (not readmitted) and class 1(readmitted) are as follows:

Instances with 0 class	**Instances with 1 class**
90409	11357

As the dataset is highly imbalanced, overall accuracy is not a good evaluation measure. Hence to evaluate the performance of a classifier we will be using kappa statistic and F-measure. Kappa statistic is a good measure as the value indicates the agreement between two raters who each classify *N* items into *C* mutually exclusive categories. F-measure also serves as good measure as it combines precision and recall, it is the harmonic mean of precision and recall.

Methods:

The dataset has been split into 80% sample and 20% sample. The 80% sample is used for training and 20% sample is used for testing in all of these methods.

Random Forest:

The data set is dominated by categorical values. The decision tree is one of the good classifier for this kind of dataset. But issue with using one decision tree is that it might result in high variance thus overfit and the performance might be poor. Hence we can use ensemble of decision trees (Random forests) to overcome this issue. By using random forest, the variance is reduced.

We can see that data set is imbalanced. Using Random Forest directly on the test data set, without any optimizations, gives the following statistics. Though the overall performance looks good there is a drastic difference between the results for the individual classes. The overall result is dominated by the class 0 instances hence it appears good. But if we have a test set dominated by class 1 instances then the performance of the model would be very poor.

Number of features to be considered in each tree of the Random Forest is obtained through cross validation.

Imbalanced training data:

Train:

Overall accuracy is 99.3109 %

Kappa statistic is 0.9644

	Classified as 0	**Classified as 1**
Actual 0	72305	0
Actual 1	561	8547

	Precision	**Recall**	**F-Measure**
Class 0	0.992	1.000	0.996
Class 1	1.000	0.938	0.968
Weighted Avg.	0.993	0.993	0.993

Test:

Overall accuracy is 88.90%

Kappa statistic is 0.0438

	Classified as 0	**Classified as 1**
Actual 0	18021	83
Actual 1	2175	74

	Precision	**Recall**	**F-Measure**
Class 0	0.892	0.996	0.941
Class 1	0.482	0.030	0.056
Weighted Avg.	0.847	0.889	0.843

Though the overall statistics looks good for test data, the F-measure for class 1 is very low and also the value of Kappa statistic is very less. This indicates that performance of the model is very poor compared to the original observed values with consideration of the chance random agreement. This is primarily due to the imbalance in the dataset. If there exists a classifier which always predicts class 0 for all the instances still it will have a good accuracy.

In training phase any classifier would aim to achieve high accuracy. The training data set is dominated by class 0 instances. Hence to achieve high overall accuracy the classifier focuses on class 0 instance and ignores the class 1 instance completely. This is evident from the class 1 F-measure and kappa statistic of the model.

Balanced training data:

To balance the train dataset, we need to resample the training dataset to increase class 1 instances and then train it using random forest. After training the Random forest using the sampled data set following statistics are observed.

Train:

Overall accuracy is 99.9816 %

Kappa statistic is 0.9996

	Classified as 0	**Classified as 1**
Actual 0	47024	2
Actual 1	13	34373

	Precision	**Recall**	**F-Measure**
Class 0	1.000	1.000	1.000
Class 1	1.000	1.000	1.000
Weighted Avg.	1.000	1.000	1.000

Test:

Overall accuracy is 83.6388 %

Kappa statistic is 0.1322

	Classified as 0	**Classified as 1**
Actual 0	16544	1560
Actual 1	1770	479

	Precision	**Recall**	**F-Measure**
Class 0	0.903	0.914	0.909
Class 1	0.235	0.213	0.223
Weighted Avg.	0.829	0.836	0.833

After oversampling the minority instances we can see that F-measure of Class 1 has increased significantly. Which means the classifier is doing better in predicting the class 1 instances even though there is a slight decrease in the class 0 F-measure, we can see that kappa static has increased significantly which indicates that after resampling the training data set the classifier performance has improved.

Random Forest with ADABoost:

Regular Random Forest uses Bagging (Bootstrap aggregating) to randomly select instances and it also randomly selects subset of the features. But we use Boosting (AdaBoost) to give higher weights to the misclassified instances while building the random trees. This will help in getting the values for the class-1 right as those will get higher weights after repeated misclassifications. In the final we have list of classifiers (multiple Random Trees) and we take majority vote from them to predict the final classification. By using this we get the following statistics.

Train:

Overall accuracy is 71.1729 %

Kappa statistic is 0.3109

	Classified as 0	**Classified as 1**
Actual 0	49057	23253
Actual 1	216	8887

	Precision	**Recall**	**F-Measure**
Class 0	0.996	0.678	0.807
Class 1	0.277	0.976	0.431
Weighted Avg.	0.915	0.712	0.765

Test:

Overall accuracy is 61.8582 %

Kappa statistic is 0.1073

	Classified as 0	**Classified as 1**
Actual 0	11207	6892
Actual 1	871	1383

	Precision	**Recall**	**F-Measure**
Class 0	0.928	0.619	0.743
Class 1	0.167	0.614	0.263
Weighted Avg.	0.844	0.619	0.69

These statistics show the improved Recall and F- measure in comparison to

K Nearest Neighbors

Since the imbalanced dataset is biased towards the majority class (class 0), here we are oversampling the minority class (class 1), balancing the dataset, normalizing the feature values and using the ensemble technique bagging as classifier for the model.   Also by doing cross validation, here we found out K to be 10 and following are the metrics observed.

Overall accuracy is 83.6388 %

Kappa statistic is 0.1107

	Classified as 0	**Classified as 1**
Actual 0	15273	2831
Actual 1	1588	661

	Precision	Recall	F-Measure
Class 0	0.906	0.844	0.874
Class 1	0.189	0.294	0.230
Weighted Avg.	0.827	0.783	0.803

The balanced data set using KNN as classifier and the bagging ensemble learning technique, increases the F-Measure when compared to the imbalanced data set but KNN classifier does not perform on this data well because it uses the distance measure for classification and works best for discrete valued attributes. In our dataset we have many categorical attributes.

Conclusion

We did all our experiments by resampling training data set to balance the class distribution. Once we tried to resample the given the given data set and then split the train and test data. This approach gave test accuracy of around 97% in random forest. But we decided this is not the right approach to test the model performance because when we oversample the class 1 instances and split the data to train and test. There is high chance that most of class 1 instances are present in both train and test which results in high accuracy so we divided dataset into train and test sets and oversampled only the training data set.

The choice of the good model also depends on the cost of misclassifying class 0 and class 1 so we tried different models. We evaluated Random Forest, Random Forest with Boosting and KNN with Bagging for the given data set. Random Forest performs better of all three as it reduces the variance using multiple trees and gives equal importance to both the classes in oversampled data. Since the training data set is resampled giving equal importance to both 0 and 1 instances the random forest works well. Boosting with Random Forest doesn’t perform well as more 0s are classified as 1 falsely. KNN doesn’t perform better because the data set is dominated by categorical attributes and there is no measure of distance for the categorical values.

References

[1] Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, “Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records,” BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014.

[2] Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records by Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore