REPORT

BEST CLASSIFIER?

ABSTRACT

This report explores two inquiries. To start with, for a given choice of datasets, would we be able to say what is the best classifier in terms of good predictions? What amount does the appropriate response rely upon the choice of datasets? How much does the answer depend on our computational constraints? We examine this inquiry utilizing datasets from the UCI repository. Second, are there some other potential routes through which we can improve the performance, considering the computational efficiency as well. Also, after having a brief idea of some of the machine learning concepts, have you tried your knowledge to some other datasets.

INTRODUCTION

Supervised learning, one of the categories of Machine learning, is further classified as Classification or Regression problem depending upon the dependent variable: categorical or discrete. Before diving into building models, it is necessary to know answers for rising questions: what our data looks like, are there any missing values, does it require scaling/normalization, what are the types of features and are all required in building models and lot more. Our very first step is to load the data sets in a format easy to interpret. We used pandas for reading and manipulation of data, since it provides hands on features to load data in tabular format (dataframe) and it has built-in methods to summarize the data set. We even checked for class imbalance by printing the frequency of each class for each data set, required for classification. This helps in choosing appropriate metrics for evaluation.

We made different functions for reading, preprocessing by applying appropriate techniques and splitting the data and process all the files the same way maintaining the consistency through out the project

Now that we have clean the data, we can feed it to defined classifiers. We can train on the training data and evaluate the model on the testing data. But, this will lead to overfitting as we are tuning the hyper parameters based on the testing data. Instead, better approach is to use a left out training data called validation data and evaluate model on it. Finally, predict the model on test data. This way we can avoid overfitting the model. We have used hyperparameters selection technique like GridSearchCV and RandomSearchCV.

Once we have best parameters for each model, finally we can fit the model on the best hyperparameters and check for the accuracy. Here, we deal with different metrics based on the application.

METHODOLOGY & EXPERIMENTAL RESULTS

This section deals with common strategy applied to both classification problems.
Data loading: We proposed a common method for classification task to read the whole dataset. **Reprocessing**: In this step, we clean the data by handling missing values. There are many techniques to do so like use the mean value, most frequent value or any value very small or high can be used to fill missing values. Also, there can be columns with categorical features and are required to be in numeric format for most of the Machine learning, though so algorithms like decision tree can handle them. So, we have used label encoding to convert to numeric values. **Splitting**: In this method, we split the dataset in training and testing.

EXPERIMENTS:

After reading the data, it is necessary to check for the class imbalance especially for classification, that will provide an overview of what metrics to evaluate on. For applications where cost of false negatives is high, recall would be a better metrics to choose. While, in scenarios where cost of false positive is high, precision is the good metrics to evaluate. But, if we need to seek a balance between Precision and Recall, then it is better to go with f1 score. Following section describes methodologies used for each classification dataset including evaluation metrics.

- Diabetic Retinopathy: In this data set, it is important that model predicts the true positives accurately, it is not acceptable that any person with daibetic retinopathy is not predicted. Hence, we used recall as an evaluation metric for this data set.
- Default of credit card clients: It is obvious that there will be a very few frauds compared to normal transaction. Hence, this data set is imbalanced. We used recall for evaluation since, it is more important to predict frauds correctly.
- Breast Cancer Wisconsin: Again an imbalanced data set and also cost of true negatives is high so we have used recall. This data set contains '?' as values for a column and so we replaced that with the most frequent value. Also, the target values where 2 and 4 and so we converted to 0 and 1 respectively.
- Australian credit approval: This data set is also imbalanced. Here, cost of false positives is high and so we have used precision as an evaluation metric.
- German credit data: Same as the default credit card data set mentioned above, we used recall.
- Steel Plates Faults: This data set also contains imbalanced class. We have used f1 score as an evaluation metric since it depends on both recall and precision.
- Yeast: This is multi-class classification problem. Classes are imbalanced and so used f1 score for evaluation. This data set also contains categorical value and so we applied label encoding.
- Thoracic Surgery Data: Columns contains categorical data and so we applied label encoding for the numeric conversion. Due to class imbalance, we used f1 score for the evaluation.
- Seismic-Bumps: Here, also we have categorical non numeric value and so we applied label encoding. Used f1 score for evaluation.

CONCLUSION:

After running models on nine different datasets, we can conclude that we cannot give the title of best classifier to any one model. Different techniques respond differently to different datasets. A number of things like size of dataset, number of features and type of values in the data. Also, we could see that there is a clear trade off between accuracy and computational complexities and time taken to train the model. thorough analysis, for Classification we can say that ensembles have performed well. AdaBoost performed much better on an average against rest of their competitors. Multiple estimators in ensembles kicks out the uncertainty and arbitrariness of models like decision tree. Also, Logistic Regression gives surprisingly good results. It's performance of significantly increases after hyper-parameter search. It works best when working with binary data and also takes significantly less time to train compared to ensembles. One more observation from our analysis is that Logistic Regression didn't over-fit. Support Vector Machines give decent results but take a lot of time when number of features are more. This can be a very difficult task for an amateur in the field. Gaussian Naive Bayes result were very bad. The biggest reason can be the assumptions of independence upon which it works. But if those assumptions hold true, it can give very good results with very less training data.

So, if a colleague of mine were to download similar datasets like the ones in this study, AdaBoost is definitely a go to. With exhaustive search it can give great result. In case of lack of good computational resources Logistic Regression is the first one a person should try. At last we can conclude that there are pros and cons of using any technique. With better understanding of the data, one can figure out which model could work best.

OVERVIEW OF THE PROJECT CODE AND DATA:

Please find all information in README.txt file in the project folder. More Plots and results of experiments are present in the Plotting_Graph. Exhaustive result and best model details are in XLS files in the project folder.

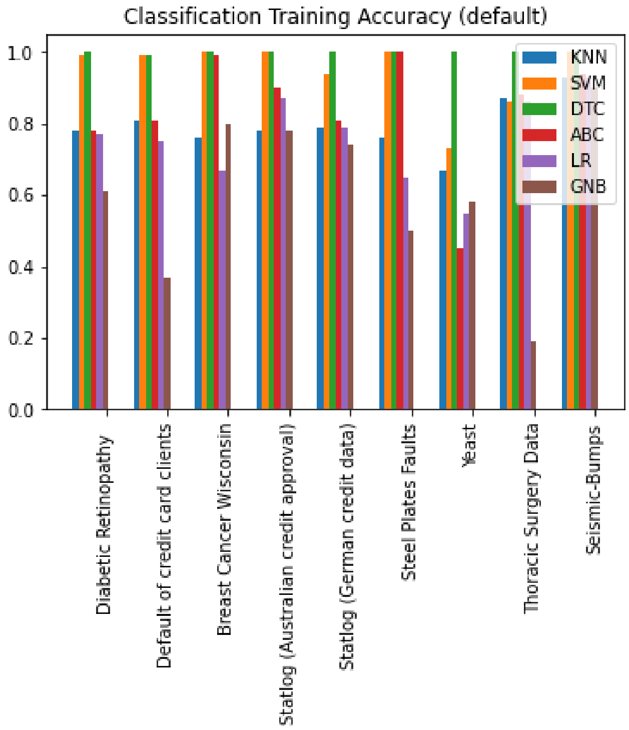

Figure 1: Classification on training accuracy (default)

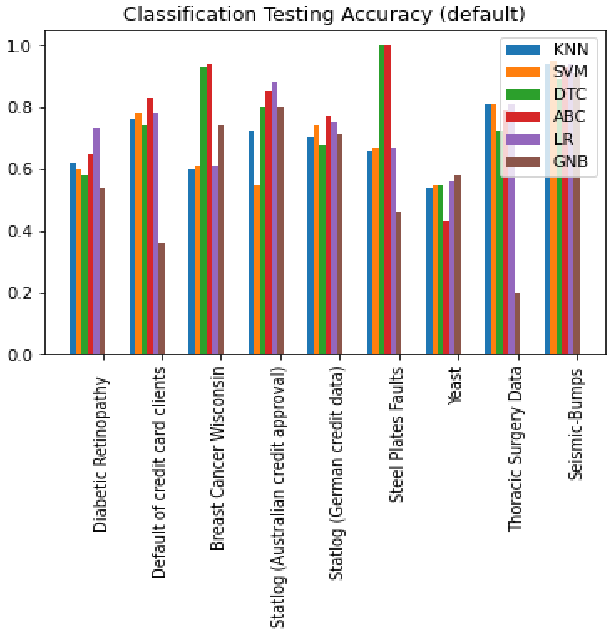

Figure 2: Classification on testing accuracy (default)