A comparative study of Classification Models in Machine Learning

Pravesh Gupta　Manjot Kaur Dherdi　Konstantin Chemodanov　Manish Yadav

Abstract

This report aims to answer the following questions. Which classification model is the best choice, given a dataset? How different models behave when trained on different datasets? For the project, we have trained all the models on few datasets selected from UCI repository and the choice of 'best model' for dataset is made by evaluating their performances in making predictions. Comparisons among models are made based on ROC AUC scores, training and testing accuracy while also keeping into account the time taken to train the model. Observations are made about the behavior of a model with different types of datasets. The need for data pre-processing is assessed and required feature transformations are implemented.

Introduction

The classification task is one of the classical machine learning problems and we decided to investigate binary and multi-class classification to acquire practical knowledge of different machine learning algorithms. Next step on our ML journey was a search of datasets suitable for our study. We sifted through hundreds of datasets available on the website of University of California, Irvine. Finally, we selected 7 datasets from UCI [1] and 1 dataset from city of Montreal open data portal [6]. Montreal crime dataset is an example of the data from our immediate environment collected since 2015 and representing criminal situation while protecting private information, criminal events are localized up to the nearest street intersection.

3 of the datasets are multiclass datasets, the rest are binary. After datasets inspection we decided that some preprocessing is required, namely categorical data encoding and parsing date values. Additional dataset investigation was done to determine if the model contains outliers.

We agreed on using grid for hyperparameters search and cross validation strategy. We decided to calculate ROC AUC score and plot ROC AUC curve for visual interpretation. Besides the accuracy, we also opted to calculate training and execution time for different classification models. We considered to investigate the need of feature transformations for successful implementation of classification algorithms and confusion matrix interpretations for some parts of overall analysis. Our ultimate goal of the project is comparison of ML models based on their performance and determining if some models work better with specific datasets.

Finally, after careful review of the goals and techniques we started the implementation of machine learning classification algorithms. We randomly divided 8 models between 4 team members, and everybody had a chance to implement their 2 models in a best possible way. Everyone used all 8 datasets for model training and testing. After many days and nights of coding, collaborations, discussions, and troubleshooting we came up with some observations, calculations, and results.

Methodology and Experimental Results

Classification of data is one of the most sought problems in machine learning. We, group of 4 people, selected 8 classification machine learning models to apply them on the varied set of datasets to find out the variations of application of each model on each of the varied dataset. The methodology is described in 3 parts. First part will explain the selection of inputs for analysis. Second part will explain the analysis itself. Third part will explain the outputs of the analysis.

As inputs to analysis, 8 classification models were chosen based on our knowledge of models and the ones which are mostly sought after. Each 2 of the 8 classification model classifiers were assigned to each of the team member to investigate and implement.

#	Classifier Name	Assignee
1	K-Nearest Neighbour (K-Means)	Pravesh Gupta
2	Support Vector Machine	Manjot Kaur

3	Decision Tree	Manish Yadav
4	Random Forest Tree	Konstantin Chemodanov
5	AdaBoost	Konstantin Chemodanov
6	Logistics Regression	Pravesh Gupta
7	Gaussian Naïve Bayes	Manjot Kaur
8	Neural Network	Manish Yadav

Fig. 1. List of Classifiers

We selected 8 datasets based on dataset type variations.

#	Dataset Name	Characteristics
1	Occupancy Detection[2]	Multivariate, Time-Series
2	Activities of Daily Living Recognition Using Binary Sensors[4]	Multivariate, Sequential, Time-Series, Multiclass
3	BitcoinHeist Ransomware Address[7]	Multivariate, Time-Series
4	Bank Marketing[3]	Multivariate
5	Montreal Crime[6]	Multivariate, Time-Series, Multiclass
6	Default of Credit Card Clients[5]	Multivariate
7	Census Income	Multivariate
8	Yeast	Multivariate, Multiclass

Fig. 2. Dataset Characteristics

Each team member worked on each of the dataset with the assigned models.

Coming to analysis itself, we all followed a coherent approach in finding the best of each model on each of the dataset. We applied following steps in each dataset evaluation: -

a. Data Loading

We loaded the dataset under analysis using numpy and sometimes with panda library based whether the data is already in numbered format or is having labels. Whenever data had labels which need to be replaced to numerical values, we took help from panda library. Some of the datasets came out of the box with proper division of training and test data but mostly we divided the dataset into 80:20 ratio for training and test data after loading the required data files. Some of the datasets contain hundred of thousand of datapoints which resulted in long-running processes. Because of computational limitations on our local machines, we had to trim certain datasets as well to a limit which gives us a good analysis and does not take more than half an hour to train.

b. Data Analysis

For some of the models, we have found out the statistics of data to further analyze the type of data we are dealing with. For some of the features, feature density and outliners were calculated and analyzed using histograms and boxplots.

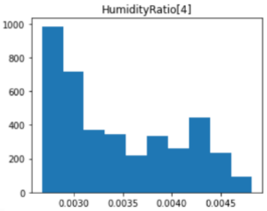


Fig. 3. Occupancy Dataset: *Temperature* Feature Values Density

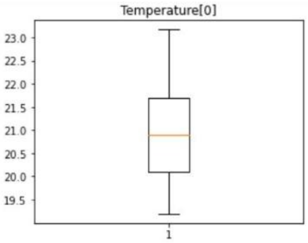


Fig. 4. Occupancy Dataset: *Temperature* Feature Box Plot

c. Model Evaluation

Since models were already divided among us, we were free to evaluate the best using each model and tweaking each of the model specific hyperparameters. We used *GridSearchCV* provided by scikit-learn library which provides an exhaustive search over the parameter spaces.

Classifier Name	Hyper Parameters
K-Means	n_neighbors
SVM	C, gamma
Decision Tree	max_features, max_depth, min_samples_split, min_samples_leaf
Random Forest	n_estimators, max_depth
AdaBoost	n_estimators
Logistics Regression	fit_intercept, solver, max_iter, penalty, C
Gaussian Naïve Bayes	None
Neural Network	batch_size, momentum, activation

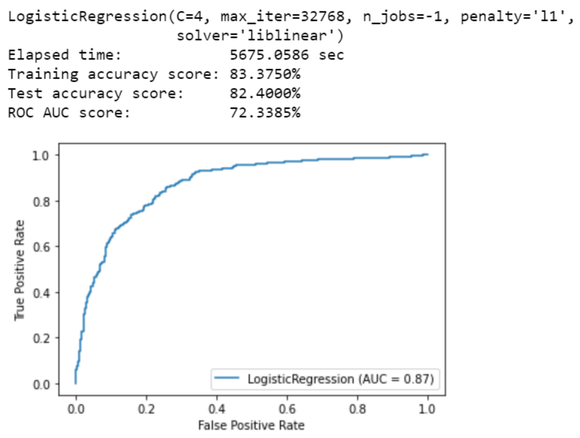
Fig. 5. Tuned Hyper Parameters for Classifiers

Since *GridSearchCV* internally implements cross-validation as well, we have good trust for the best estimator returned for issues like overfitting or underfitting. We also explored impact of dimensionality reduction and feature processing on some models. Since gaussian naïve bayes was observed to train badly, we thought it was good idea to study these techniques on this model.

d. Model Evaluation Results

Most of the meaning of the machine learning comes from its model evaluation results. In our case, we collected metrics for each of the model analysis on each dataset to be used for the comparison at the end of the evaluations. For each analysis, we extracted following metrics:

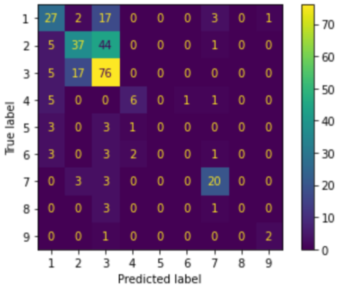
- *Elapsed Time*: This represents the time required to find the best parameters in a grid search of parameter space so that the model gives best accuracy. However it is not time to train individual model but the total time of whole parameter space search evaluating models arising out of all combinations, we think that it indeed is valuable since parameter search is also one of the main tasks for best model search and hence we proceeded to use it for comparison purposes. Lower the time taken to search for parameters, better the model from the performance point of view.
- *Training Accuracy Score*: For best estimator, the accuracy score over training data. Higher the training accuracy score the better is the model for the input data and hence better at classifying the current known labels.
- *Test Accuracy Score*: For best estimator, the accuracy score over test data. Higher the test accuracy score over, the better is the model on new set of data and hence better at future predictions.
- *ROC AUC score*: This score helped us evaluate the True Positive Rate and False Positive Rate comparisons. Mostly, higher the ROC AUC score, better the model for predictions and the same has been applied in comparison analysis.



```
LogisticRegression(C=4, max_iter=32768, n_jobs=-1, penalty='l1',
                   solver='liblinear')
Elapsed time:            5675.0586 sec
Training accuracy score: 83.3750%
Test accuracy score:     82.4000%
ROC AUC score:           72.3385%
```

Fig. 6. LR on Census Dataset: ROC curve

- *Confusion Matrix*: For some of the models, we tried to analyze confusion matrix as well for the incorrect class assignment trends.



True label \ Predicted label	1	2	3	4	5	6	7	8	9
1	27	2	17	0	0	0	3	0	1
2	5	37	44	0	0	0	1	0	0
3	5	17	76	0	0	0	0	0	0
4	5	0	0	6	0	1	1	0	0
5	3	0	3	1	0	0	0	0	0
6	3	0	3	2	0	0	1	0	0
7	0	3	3	0	0	0	20	0	0
8	0	0	3	0	0	0	1	0	0
9	0	0	1	0	0	0	0	0	2

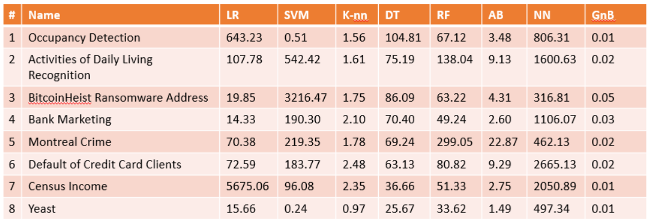
Fig. 7. LR on Yeast dataset: Confusion Matrix

Observations and Interpretations

After collecting and comparing all metrics and applying different machine learning concepts, we are able to make certain observations and make some interpretations out of those.

1. Time Required for Parameter Space Search

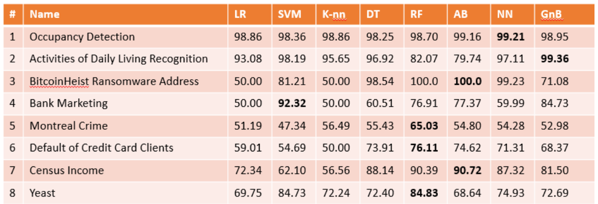
We are able to see that Gaussian Naïve Bayes, K-Means and AdaBoost are most efficient in terms of time required while Neural Network take a lot of time on average. Decision Tree and Random Forest seems stable in terms of variance in required time while Support Vector Machine time requirement is quite varied. This analysis can be taken as a reference to the computational requirements for different classifiers.



#	Name	LR	SVM	K-nn	DT	RF	AB	NN	GnB
1	Occupancy Detection	643.23	0.51	1.56	104.81	67.12	3.48	806.31	0.01
2	Activities of Daily Living Recognition	107.78	542.42	1.61	75.19	138.04	9.13	1600.63	0.02
3	BitcoinHeist Ransomware Address	19.85	3216.47	1.75	86.09	63.22	4.31	316.81	0.05
4	Bank Marketing	14.33	190.30	2.10	70.40	49.24	2.60	1106.07	0.03
5	Montreal Crime	70.38	219.35	1.78	69.24	299.05	22.87	462.13	0.02
6	Default of Credit Card Clients	72.59	183.77	2.48	63.13	80.82	9.29	2665.13	0.02
7	Census Income	5675.06	96.08	2.35	36.66	51.33	2.75	2050.89	0.01
8	Yeast	15.66	0.24	0.97	25.67	33.62	1.49	497.34	0.01

Fig. 8. *Elapsed Time on each classifier for each dataset*

2. *Overall Accuracy of Classification Model*



#	Name	LR	SVM	K-nn	DT	RF	AB	NN	GnB
1	Occupancy Detection	98.86	98.36	98.86	98.25	98.70	99.16	**99.21**	98.95
2	Activities of Daily Living Recognition	93.08	98.19	95.65	96.92	82.07	79.74	97.11	**99.36**
3	BitcoinHeist Ransomware Address	50.00	81.21	50.00	98.54	100.0	**100.0**	99.23	71.08
4	Bank Marketing	50.00	**92.32**	50.00	60.51	76.91	77.37	59.99	84.73
5	Montreal Crime	51.19	47.34	56.49	55.43	**65.03**	54.80	54.28	52.98
6	Default of Credit Card Clients	59.01	54.69	50.00	73.91	**76.11**	74.62	71.31	68.37
7	Census Income	72.34	62.10	56.56	88.14	90.39	**90.72**	87.32	81.50
8	Yeast	69.75	84.73	72.24	72.40	**84.83**	68.64	74.93	72.69

Fig. 9. *ROC AUC score of each classifier for each dataset*

As we can see from the comparison table in Fig. 9 that Random Forest Classifier worked best for most of the datasets followed by AdaBoost. In our study, K-Means and Logistic Regression classifiers could not product best ROC AUC score for any of the dataset. One more observation is that Logistics Regression and K-Means ROC AUC score were similar with each other and quite comparable.

3. *Feature Processing and Dimensionality Reduction*

Three techniques were employed to study impact of data pre – processing on models like SVM and GNB. For most datasets, the results resembled to that of unprocessed data and same trend was seen on all three techniques. Following observations were made.

- *MinMaxScaler*: It has given almost similar results as before preprocessing.
- *Principal Component Analysis*: It has improved training accuracy for most datasets by little margin, but a trend was seen in testing accuracy which is very low as compared to unprocessed data and training data. In our opinion this can be due to overfitting.
- *Variance Threshold*: It provides results like unprocessed data in most cases with improvements by little margin for some datasets.

4. *Effect of standardization on Decision tree and Neural network*

While there was negligible effect on AUC for decision tree ROC, it was seen that after standardization of input data neural network worked significantly better in terms of AUC of ROC in all the datasets.

Conclusions

While there is no one single number that can be used to compare classifiers but if performance on average is considered then area of the ROC curve can be a good metric. How to make use of ROC depends on the requirements. There are cases when we want to maximize the TPR, like life critical cases of predicting cancer, in these cases a negative being predicted as positive is of less concern. On the other hand, there are cases when we want to maximize TPR but with a limitation on FPR. While those limits are decided by individual/organizations and circumstances, average performance can be measured by AUC of ROC. In datasets we used and looking from the stats perspective on average Random Forest seems to outperform all other classifiers. Pre-processing of data may not help in all the cases and even can increase running time in many cases. So, pre-processing should be done with specific goals.

We believe that we have explored most of the aims of this study and were able to find observations and interpretations using comparative modelling and different techniques. The study can be extended further to other machine learning concepts and specifics and can be used a base for further exploration.

References

[1] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Luis M. Candanedo, Véronique Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energy and Buildings. Volume 112, 15 January 2016, Pages 28-39.

[3] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

[4] Ordóñez, F.J.; de Toledo, P.; Sanchis, A. Activity Recognition Using Hybrid Generative/Discriminative Models on Home Environments Using Binary Sensors. Sensors 2013, 13, 5460-5477.

[5] Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.

[6] Ville de Montréal (2020). Criminal acts. [https://donnees.montreal.ca/ville-de-montreal/actes-criminels].

[7] Akcora, Cuneyt & Li, Yitao & Gel, Yulia & Kantarcioglu, Murat. (2019). BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain.