Research Project Report

Soh Ornella Lucresse

May 12, 2020

0.1 Introduction

Since the end of twenty century with a significant increase in technology and the availability of data, Machine Learning has become ubiquitous in addressing the various types of problems in the real world. However, it turns out that for any classification or regression problem, we can use different Machine Learning techniques to address them. Then, the challenge we always encounter is which technique is the most appropriate to solve that particular problem?
In order to address this challenge, a comparative study of machine learning algorithms can be performed to evaluate the performance of these different techniques throughout real and simulated datasets.
However, In this project, we are working on classification problem which aims to compare three different Machine Learning algorithms: Logistic Regression (LR), K Nearest Neighbor (KNN) and Support Vector Machine (SVM), on a real and simulated dataset.
This work is divided into three parts where in the first part we will be working with real datasets while in the second part, we will simulate a dataset from the exponential distribution. Then, we will end with the discussion of our results. In the first and second parts, we will start with data description, followed by data visualization which will allow us to understand the distribution before starting the preprocessing, and then, build classifiers with logistic regression, K Nearest Neighbor and Support Vector Machine train on the given dataset. Finally compare these three methods base on the accuracy, training, and test time.

0.2 Real datasets

0.2.1 Data Description

The Statlog Shuttle, from UCI archive [3], is the dataset used for this project. It contains nine attributes all of which are numerical. The first column refers to time and the last column is referred to classes that contain seven categories, Rad Flow, Fpv Close, Fpv Open, High, Bypass, Bpv Close, and Bpv Open labeled from 1 to 7. The dataset consists of two files, namely *shuttle.trn.Z* and *shuttle.tst* where the training and test set contains 43500 and 14500 examples respectively. The visualization of the training data led us to the following figure describing the distribution over classes.

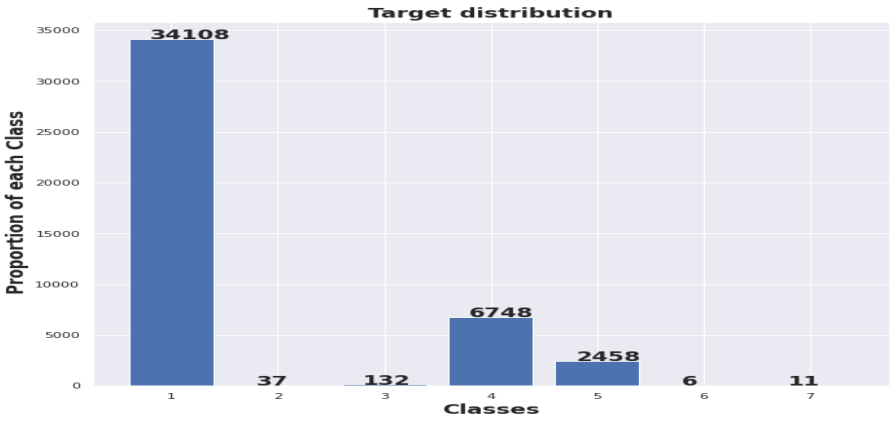

Figure 1: Data distribution

0.2.2 Preprocessing

1. **Outliers**
   An outlier is an observation that lies outside the overall pattern of distribution. It considerably affects the performance of a model. In order to check for outliers in the data, we plot the histogram of every variable as shown in figure 2. We realized that there is a lot of outliers in variable1, variable3, and variable5. To confirm the effective presence of outliers, we visualize the scatter plot of such variables with respect to the first variable that refers to time as shown in figure 3.

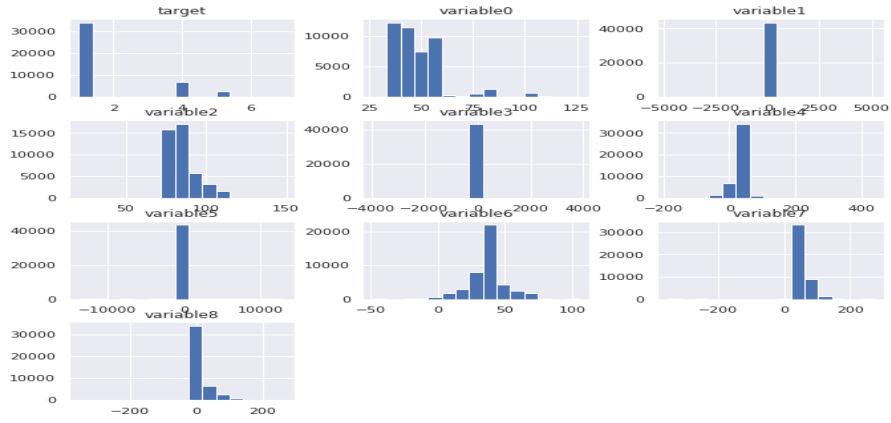

Figure 2: Data histogram

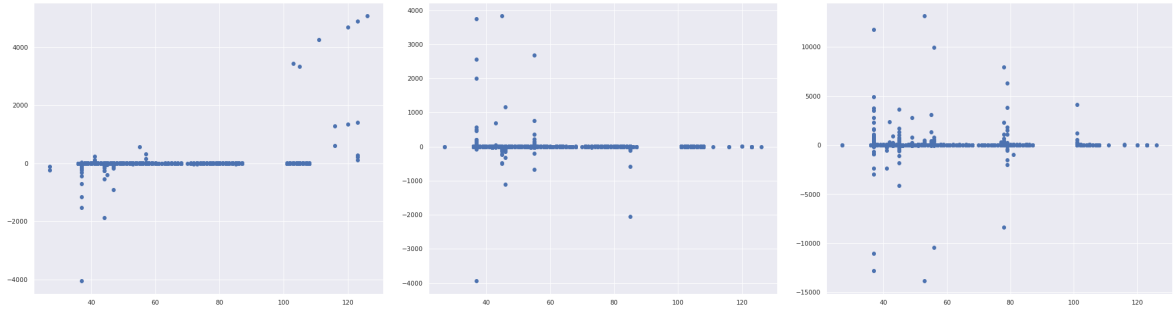

Figure 3: Scatter plot of variable1, variable2 and variable3

2. **Imbalanced data**
   Knowing that the problem of imbalanced data has a significant impact when training a model, we first check the proportion over each class as described in table 1 and realized that some classes such as 3, 2, 6, and 7 have a proportion less than 0.005. Then, we decided to remove these classes because they are not significant.

Classes	1	4	5	3	2	7	6
Proportion	0.7841	0.1551	0.0565	0.0003	0.00009	0.00003	0.00001

Table 1: Proportion over classes

   Finally we appy random over sampling to fit the imbalanced data in the remaining classes.

3. **Data scaling**
   In order to standardize the data, we applied data scaling also known as data normalization is the method used to standardize the range of features of data since the range of values of data may vary widely.

0.2.3 Method

Different methods have been used to compare Machine Learning algorithms which give differents significant results. In this section, we will take turns explaining them.

1. **Logistic regression**
   Logistic Regression is one of the important technique in machine learning used for classification problem. It underliesd on the key concept of logits which is a transformation of an odds ratio by applying the logarithm function [1]. It turns out that this technique is more robust for binary classification problem. However, to build our logistic classifier, we considered the following hyperparameters:

   - **Penality (c)** is set to 10. The intuition of choosing this value is that we don't to apply high regularization into parameters.
   - **Random_state** is set to 0.

   The remain of the hyparameers of the classifier are set by default. Then, we got the following classifier.

   LogisticRegression(C=10, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='auto', n_jobs=None, penalty='l2', random_state=0, solver='lbfgs', tol=0.0001, verbose=0, warm_start=False)

Table 2 summarizes the accuracy, training and testing time of the above classifier after training and make prediction.

Trainig time	Testing time	Accuracy
5.431	0.0033	97.413%

Table 2: Logistic Regression summary

2. **K Nearest Neighbor (KNN)**
The K-nearest neighbor is the simplest method for classification problems. The hyperparameter k refers to the number of nearest neighbors. The intuition behind this method is to find the K nearest neighbor for each point based on the appropriate metric. They are many metric used in KNN such as **Euclidien distance, Minkowski distance, etc** and the most common used seem to be **Euclidien distance** because its eases to use and finds dissimilarities between two points [2]. However, before build the KNN classifier, let us first define all the hyperparameter.

   - **Metric:** minkowski
   - **K:** This important hyperpareter have been set using **Elbow** method which aim to compute the error as a function of K. From that, we found the optimal K to be 2.

   The remain of the hyparameers of the classifier are set by default. Then, we got the following classifier.

   KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski', metric_params=None, n_jobs=None, n_neighbors=2, p=2, weights='uniform')

   Table 3 summarizes the accuracy, training and testing time of the above classifier after training and make prediction.

Trainig time	Testing time	Accuracy
6.069	1.262	99.979%

Table 3: KNN summary

3. **Support Vector Machine**
The standard Support Vector Machines are designed for binary classification. To solve the problem of multi-class classification using SVM, we commonly decompose that multi-class into several binary classification [4].

   The following hyperparameter are used to build the SVM classifier:

   - **Penalty (C):** This hyperparameter is used to regularize the error term. In this framework, we chose $C = 2$
   - **Kenel function:** The kernel used is **rbf**.

   The remain of the hyparameers of the classifier are set by default. Then, we got the following classifier.

   SVC(C=2, break_ties=False, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='
   degree=3, gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

   Table 4 summarizes the accuracy, training and testing time of the above classifier after training and make prediction.

Trainig time	Testing time	Accuracy
5.623	0.594	99.979%

Table 4: SVM summary

0.3 Simulated data

In this section, instead of working on real datasets, a dataset that contains 40000 examples have been simulated based on the fact that the left and right distributions are exponential with parameters λ_0 and λ_1 respectively. In order to make the simulated dataset more reliable, we decided to add some noise sample from a normal

distribution. The visualization of the training data leaded us to the following figure describing the distribution over classes.

Curious to see the impact of the parameters λ_0 and λ_1 in the result, we decided to vary these parameters to

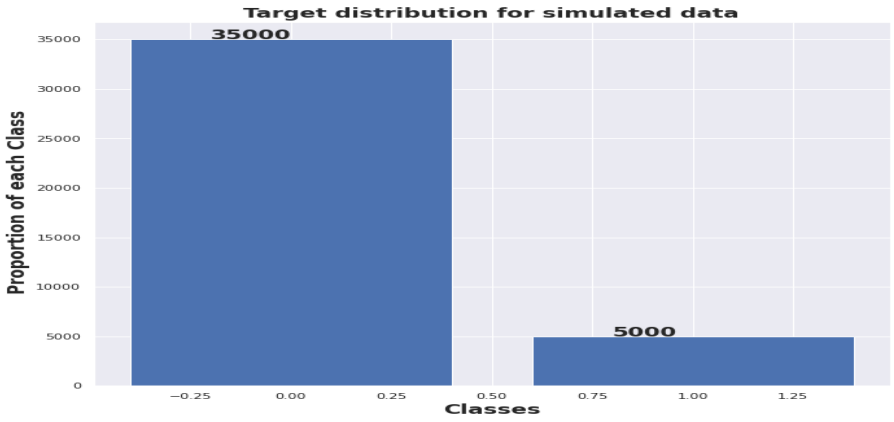

Figure 4: Simulated data distribution

check if the accuracy will change or not. Then, we realized that for the Logistic and Support Vector Machine classifier, the accuracy varies slightly while in K Nearest Neighbor, it varies considerably.

0.4 Discussion

Significant results have been obtained from the three Differents classifiers on real and simulated datasets. In this section, we compare those classifiers in terms of accuracy, training, and testing time.

1. **Real dataset** By applying these classifiers on the real dataset, we realize that Logistic Regression performs well than K Nearest Neighbor and Support Vector Machine according to the training, and testing time. While Support Vector Machine and K Nearest Neighbor have the same accuracy which outperforms logistic classifier.
   The table below summarizes the different results obtained on the real dataset.

Classifier	Trainig time	Testing time	Accuracy
LR	5.431	0.0033	97.413%
KNN	6.069	1.262	99.979%
SVM	5.623	0.594	99.979%

Table 5: Results summary on real dataset

2. **Simulated dataset** Using the simulated dataset, it turns out from the summarized tables below that for any given λ_0 and λ_1, K Nearest Neighbor classifier outperforms Logistic and Support Vector Machine classifiers in term of accuracy, while in term of training and testing time, Logistic regression algorithm perfoms well compared to the others classifiers.

Classifier	Train time	Test time	Acc
LR	0.043	0.0006	50.77%
KNN	0.053	0.485	91.47%
SVM	1.545	0.298	50.90%

Table 6: For $\lambda_0 = 1$ and $\lambda_1 = 1$

Classifier	Train time	Test time	Acc
LR	0.042	0.0006	50.59%
KNN	0.055	0.49	91.73%
SVM	1.54	0.295	49.6%

Table 7: For $\lambda_0 = 120$ and $\lambda_1 = 0.9$

Classifier	Train time	Test time	Acc
LR	0.044	0.0008	49.82%
KNN	0.052	0.46	92.29%
SVM	1.550	0.29	49.95%

Table 8: For $\lambda_0 = 0.06$ and $\lambda_1 = 0.02$

Classifier	Train time	Test time	Acc
LR	0.046	0.0006	50.08%
KNN	0.055	0.49	91.74%
SVM	1.59	0.28	50.50%

Table 9: For $\lambda_0 = 4$ and $\lambda_1 = 12$

0.5 Conclusion

Through this project, we were able to build three different classifiers for the classification problem of the shuttle dataset, such as Logistic Regression, Nearest Neighbor, and Support Vector Machine. We also apply these classifiers into a simulated dataset sampled from an exponential distribution. Then, we used a comparative approach to find the best classification system for the given dataset, based on the training time, test time, and accuracy. Finally, for the real dataset, we realized that the accuracy of these three classifiers was better than 97%, which allows us to conclude that choosing a different classifier for this particular dataset is therefore not a good idea, as it may not outperform these classifiers we built so far.

Bibliography

[1] Peng, Chao-Ying Joanne, Kuk Lida Lee, and Gary M. Ingersoll. "An introduction to logistic regression analysis and reporting." The journal of educational research 96.1 (2002): 3-14

[2] Weinberger, Kilian Q., John Blitzer, and Lawrence K. Saul. "Distance metric learning for large margin nearest neighbor classification." Advances in neural information processing systems. 2006.

[3] Newman, D.J., Asuncion, A., 2007. UCI Machine Learning Repository. University of California, Irvine, Dept. of Information and Computer Sciences.

[4] Franc, Vojtech, and Václav Hlavác. "Multi-class support vector machine." Object recognition supported by user interaction for service robots. Vol. 2. IEEE, 2002.