REPORT - GROUP WORK, PROJECT SUBMISSION 2

Students:

David Kwasi Nyonyo Mensah-Gbekor(mensahgbekor@hotmail.com)
David Wonder Doe-Dekpey (wonderdoe85@yahoo.com)
Alexander Botica (alexbotica@yahoo.com)
Alexander Victor Okhuese (alexandervictor16@yahoo.com)

INTRODUCTION

According to Nargesian *et al.* (2017), feature engineering is the task of improving predictive modelling performance on a dataset by transforming its feature space. Existing approaches to automate this process rely on either transformed feature space exploration through evaluation-guided search, or explicit expansion of datasets with all transformed features followed by feature selection (Nargesian *et al.*, 2017). Such approaches incur high computational costs in runtime and memory. Browiee (2014) defines feature engineering as the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. Coming up with features is difficult, time-consuming, requires expert knowledge. In this work we aim to describe some feature engineering techniques used in the second task of work group as well as some cross-validation techniques.

LITERATURE REVIEW

Feature Engineering Techniques

In Machine Learning, the large number of attributes indicates the complexity of a problem. To solve complex classification problem, good features plays a crucial role, which have a higher predictive power. Engineering good features set is prerequisite to achieve high accuracy in classifying objects and prediction. Feature Engineering is the foundation to learning algorithms. It is the process of using domain knowledge of the data to create features that make learning algorithms work (Berend and Farkas, 2010). As stated by Rawat and Khemchandani (2017), in order to work predictive/classified

learning algorithms for underlying problems, inputs must be transformed into such format so that algorithm understands it and gives the accurate class to which an entity belongs and future prediction.

IMPUTATION

One of the basic principles for using an imputation procedure is to determine what is a missing value (Schafer and Graham, 2002). A missing value is usually represented by an absence code - for example, it can be a symbol, a specific value or a blank, meaning that a particular value has not been observed or was simply ignored. The occurrence of such values usually makes the task of data analysis difficult. When such an event is beyond our control, it is possible to make assumptions about how these values are distributed (Schafer and Graham, 2002). Most machine learning algorithms cannot deal with this problem intrinsically, as they are designed for complete data. A number of simple approaches exist. For basic use cases, these are often enough. In this assignment we implemented two strategies to deal with large numbers of missing values. The first approach was to input zero values to not available (NAs) data points and the second one was to input average values. According to Song and Shepper (2007), this last approach has the advantage of being the simplest possible, and one that does not introduce any undue bias into the dataset.

HANDLING OUTLIERS

An outlier is defined as an observation that deviates from other observations with respect to a measure and exerts a substantial influence on data analysis (Dhakal, 2017). Outliers and influential points are sensitive to regression analysis. According to Osborne and Overbay (2004), possible effects can cause increase in error variance to reduce the power of statistical test, help violet the assumptions in the model which ultimately result a biased estimate. Many strategies are used to define a rule to remove outliers. In our case, we have chosen to remove values above and below 0.95 and 0.05 quantiles, respectively.

LOG-TRANSFORMATION

Sometimes the symmetric bell-shaped distribution often does not adequately describe the observed data. Quite often data arising in real studies are so skewed that standard statistical analyses of these data yield invalid results. Many methods have been developed to test the normality assumption of observed data. When the distribution of the continuous data is non-normal, transformations of data are applied to make the data as 'normal' as possible and, thus, increase the validity of the associated statistical analyses. The log transformation is, arguably, the most popular among the different types of transformations used to transform skewed data to approximately conform to normality, and that is the reason why we have chosen it.

SCALING

As the name suggests, feature scaling changes the scale of the feature. Sometimes it is also call feature normalization. Feature scaling is usually done individually to each feature.

One of the strategies implemented in this assignment used Min-max scaling to squeeze or stretches all feature values to be within the range of [0, 1]. Equation:

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Another strategy consists in subtract off the mean of the feature and divide by the variance. Hence, it can also be called variance scaling. The resulting scaled feature has a mean of 0 and a variance of 1. If the original feature has a Gaussian distribution, then the scaled feature does too. Equation:

$$\tilde{x} = \frac{x - \text{mean}(x)}{\text{sqrt}(\text{var}(x))}$$

Cross-validation Techniques

Cross-Validation is a statistical method of evaluating and comparing leaming algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model (Angel and Fernandez, 2015). In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated against. The basic form of cross-validation is k-fold cross-validation. Other forms of cross-validation are special cases of k-fold cross-validation or involve repeated rounds of k-fold cross-validation. In this section we will describe three types of cross-validation techniques.

K-FOLD

K-fold cross-validation is an intensive computational technique that uses all available samples as training and test samples (Duchesne and Rémillard, 2015). According to Nelson *et al.* (2017), in relation to other cross-validation methods such as Hold-out and Leave-One-Out, with k-fold we can achieve more accurate results, often superior to Leave-One-Out which in many cases is not used for requiring performance computational resources.

Given a hypothetical database containing 100 records, and defining k = 10 the database will be divided into 10 subsets where each subset will have 10 records each. After subassembly, a subset will be used to be used in the validation of the model and the remaining assemblies are used as training. The cross-validation process is then repeated K (10) times, so that each of the K subsets is used exactly once as a test for validation of the model.

LEAVE-ONE-OUT

The Leave-One-Out Cross-Validation (LOOCV) occurs in the same way as the K-Fold method with the main difference that the training is performed with n-1 data and the test with 1 of the registers only. The Leave-One-Out method defines the number of subsets equal to the number of records in the database. Hence, if the database has 100 records inside it, 100 subsets each with 1 record will be defined. After partitioning the subsets the same K-Fold process is performed, the subset B1 is used for testing and the remainder for training, in the example case there would be 99 subsets for training, so on. In accordance with Refaeilzadeh, Tang and Liu (2008), an accuracy estimate obtained using LOOCV is known to be almost unbiased but it has high

variance, leading to unreliable estimates, though it is still widely used when the available data are very rare.

HOLD-OUT

In Hold-out validation the method resembles K-Fold where ok = 2, but with a particularity, the database is divided into two parts, with one of the parts being used for training and the other part for testing, without the alternation that occurs with k-fold. This process is performed only once, unlike the K-Fold process in which the data are divided into K parts, and each part is used for both training and testing, so that all parts go through both sides. Yadav et al. (2016), stated that one advantage of the hold-out model is that the time required to learn the model is relatively smaller than the time required to learn the model using the k-fold cross validation.

NEXT STEPS

There are some implementations we shall try for the next submission. Most more focused on time series data. For de-trending, there can be benefit in identifying, modeling, and even removing trend information from the time series dataset. A study of stationary characteristics is also a possibility, because most statistical forecasting methods are based on the assumption that the time series can be rendered approximately stationary through the use of mathematical transformations. To better see patterns, we might invest in a smoothing technique to smooth out the irregular roughness to see a clearer signal. For seasonal data specially, we might smooth out the seasonality so that we can identify the trend. Though smoothing doesn't provide us with a model, it can be a good first step in describing various components of the series. One strategy would be implementing moving averages.

CONCLUSION

When we were experimenting on Feature Engineering techniques, we used momentum and a couple other indicators as features to help in the target variable prediction. Alpha Vantage which we acquired the data from already has data for the indicators so we downloaded the data and added it to the independent variables. We used SVM in the cross validation but we will decide on a better model.

REFERENCES

Angel, M. and Fernandez, L. (2015) *Cross-validation*. Available at: https://scholar.harvard.edu/files/malf/files/maluque-cross-validation_01.pdf (Accessed: 5 May 2019).

Berend, G. and Farkas, R. (2010) *SZTERGAK : Feature Engineering for Keyphrase Extraction*. Association for Computational Linguistics. Available at: http://www.michigan-proficiency-exams.com/suffix- (Accessed: 5 May 2019).

Dhakal, C. P. (2017) 'DEALING WITH OUTLIERS AND INFLUENTIAL POINTS WHILE FITTING REGRESSION', *Journal of Institute of Science and Technology*, 22(1), pp. 61–65. doi: 10.3126/jist.v22i1.17741.

Duchesne, P. and Rémillard, B. (2005) *Statistical modeling and analysis for complex data problems*. Springer.

Little, R. J. A. and Rubin, D. B. (no date) *Statistical analysis with missing data*. Available at: https://www.wiley.com/en-us/Statistical+Analysis+with+Missing+Data%2C+2nd+Edition-p-9780471183860 (Accessed: 5 May 2019).

Nargesian, F. *et al.* (2017) 'Learning Feature Engineering for Classification', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. California: International Joint Conferences on Artificial Intelligence Organization, pp. 2529–2535. doi: 10.24963/ijcai.2017/352.

Nelson Corleta Schreiber, J. *et al.* (2017) *TÉCNICAS DE VALIDAÇÃO DE DADOS PARA SISTEMAS INTELIGENTES: UMA ABORDAGEM DO SOFTWARE SDBAYES*. Available at: https://repositorio.ufsc.br/bitstream/handle/123456789/181199/102_00071.pdf?sequence=1 (Accessed: 5 May 2019).

Rawat, T. and Khemchandani, V. (2017) 'Feature Engineering (FE) Tools and Techniques for Better Classification Performance', *International Journal of Innovations in Engineering and Technology*. doi: 10.21172/ijiet.82.024.

Refaeilzadeh, P., Tang, L. and Liu, H. (2008) *C Cross-Validation*. Available at: http://leitang.net/papers/ency-cross-validation.pdf (Accessed: 5 May 2019).

Schafer, J. L. and Graham, J. W. (2002) 'Missing data: our view of the state of the art.', *Psychological methods*, 7(2), pp. 147–77. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12090408 (Accessed: 5 May 2019).

Yadav, S. and Shukla, S. (2016) 'Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification', in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. IEEE, pp. 78–83. doi: 10.1109/IACC.2016.25.

BIBLIOGRAPHY

Emre, R. (2019) Techniques of Feature Engineering for Machine Learning Available at https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114 (Accessed: 2 May 2019)

Georgios, D. (2019) Cross-Validation Available at: https://towardsdatascience.com/cross-validation-70289113a072 (Accessed: 2 May 2019)

Jason Browniee (2014) *Discover Feature Engineering, How to Engineer Features and How to Get Good at It*, *Machine Learning Mastery*. Available at: https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/ (Accessed: 5 May 2019).

Osborne, J. W. and Overbay, A. (2004). The Power of Outliers (and why researchers shouldalways check for them). Practical Assessment, Research & Evaluation, 9 (6). North Carolina State University. Retrieved from pareonline.net/getvn.asp?v=9&n=6 Accessed on 10.7.2013

Song, Q. and Shepperd, M. (2007) 'Missing Data Imputation Techniques', *International Journal of Business Intelligence and Data Mining*, 2(3), p. 261. doi: 10.1504/IJBIDM.2007.015485.