Deep Regressor Stacking for Air Ticket Prices Prediction

Everton Jose Santana
State University of Londrina (UEL)
Electrical Engineering Department
Londrina, Brazil
santana.everton@ieee.org

Saulo Martiello Mastelini
State University of Londrina (UEL)
Computer Department
Londrina, Brazil
mastelini@uel.br

Sylvio Barbon Jr.
State University of Londrina (UEL)
Computer Department
Londrina, Brazil
barbon@uel.br

ABSTRACT

Purchasing air tickets by the lowest price is a challenging task for consumers since the prices might fluctuate over time influenced by several factors. In order to support users' decision, some price prediction techniques have been developed. Considering that this problem could be solved by multi-target approaches from Machine Learning, this work proposes a novel method looking forward to obtaining an improvement in air ticket prices prediction. The method, called Deep Regressor Stacking (DRS), applies a naive deep learning methodology to reach more accurate predictions. To evaluate the contribution of the DRS, it was compared with the competence of the single-target regression and two state-of-the-art multi-target regressions (Stacked Single Target and Ensemble of Regressor Chains). All four approaches were performed based on Random Forest and Support Vector Machine algorithms over two real-life airfares datasets. After results, it was concluded DRS outperformed the other three methods, being the most indicated (most predictive) to assist air passengers in the prediction of flight ticket price.

CCS Concepts

•Information systems → Decision support systems; Data mining; •Computing methodologies → Machine learning; Model verification and validation; •Applied computing → Online shopping; •Mathematics of computing → Regression analysis; Information theory;

Keywords

Decision Support System, Multi-Target Regression, Airfare Prediction, Price Mining

1. INTRODUCTION

The International Air Transport Association's 2016 review reported that more than 3.5 billion passengers segments were flown in 2015, and by 2034, air passenger number is forecast to increase to seven billion annually [1].

SBSI 2017 June 5^{th} – 8^{th}, 2017, Lavras, Minas Gerais, Brazil

Besides the growth in the number of airplane travellers, the access to the air ticket prices became more available on the Internet, allowing the consumers to compare, more easily, prices over time and to identify some pricing tendencies, in order to choose the right moment to purchase a ticket. However, identifying the patterns in the pricing mechanism is a complex process composed of many factors such as different classes of seats in the same flight, diverse sellers, seasonality, amount of seats available and the price of other companies, which cause a high variability in the prince over time [8]. In fact, the large dispersion in airfares for seats in the same flight, and the questioning of its reasonableness, was already indicated in [15]. The prediction becomes even harder because some variables are not accessible to consumers [8, 9] and the companies are improving yield management algorithms to optimize their profits [17].

Intending to support consumers to predict price changes, some existing data-mining methods were used (Ripper, Q-learning and Time Series) to propose the stacking generalizer algorithm Hamlet [8]. Another approach was the representation of price series by marked point processes [17]. There was also the proposal of a method that includes the preferences of passengers about the number of stops in the itinerary or the specific airline to use [9], further developed to a technique called Developer-Guided Feature Selection [10].

Observing the multiple output characteristic of some datasets, and the possible mutual dependence among the outputs, Spyromitros-Xioufis et al. [14] proposed two multi-target regression methods: Stacked Single Target (SST) and Ensemble of Regressor Chains (ERC). Both techniques use target predictions as additional input variables in order to increase the prediction accuracy. Among many validation datasets, the referred work used the one presented in [9].

Motivated by SST, we propose in this paper a novel multi-target technique denominated Deep Regressor Stacking (DRS) looking forward to obtaining an improvement in air ticket prices prediction. This model could be implemented in a costumer decision support system and, consequently, avoid that the user buy a high-priced ticket in the searching date if the price is supposed to decrease in the following days.

This paper aims at comparing the performance of the single target regression (ST) and the three multi-targets (SST, ERC and DRS) approaches to predict air ticket prices, each one with the regressors Random Forest (RF) and Support Vector Machine (SVM), and evaluates which method would bring the biggest improvement to the area.

This paper is organized as follows: Section 2 exposes existing multi-target regressions and describes our new proposal.

Section 3 portrays the experimental configuration, followed by Section 4, which analyses the results. Lastly, Section 5 contains the final considerations of this paper and suggests future work.

2. MULTI-TARGET REGRESSION

2.1 Literature Review

Traditionally, multi-target problems were solved through two broad approaches: algorithms adaptation and problem transformation methods [4].

The algorithm adaptation is related to the change of some single-target regression technique to deal with multiple outputs and address the possible statistical dependence among targets. This strategy generates alteration in the original technique modelling method, like the optimization function (SVMs) [13, 20, 19], node splitting criteria (regression trees) [11], among others. Some algorithms adaptation based techniques were proposed [4], being used in diverse tasks [12, 18, 4]. Indeed, multi-output adapted algorithms have achieved satisfactory performance in prediction, bringing the advantages of internally exploring target dependence and generating a unique model to deal with all outputs. Nevertheless, algorithm adaptation methods could be more challenging because these techniques aim not only to predict the multiple targets at once but also to interpret the dependencies among outputs.

The other approach to model multi-target tasks, problem transformation, manipulates the training data in some manner, adopting well known and applied regression techniques to predict single-target problems separately. A simply derived approach is to predict each target variable independently, as a single-target (ST) problem.

In many cases ST method outperformed multi-target techniques (based both in algorithm adaptation and problem transformations) [4, 14], and was used as a performance baseline. In contrast, ST does not explore the expected targets' dependencies, so the use of a multi-target strategy should lead better results.

Some techniques were proposed in the last years to address multi-target problems as separated single output tasks, but with the exploration of inter-targets properties. Zhang et al. [20] proposed the modification of problem's input space through a virtualization procedure so that the task could be represented as a wider single-target problem. The authors used a Support Vector Regression (SVR) machine and achieved results comparable to ST strategy. Tsoumakas et al. [16] proposed the use of random linear targets combinations to explore the relations between targets values. The original feature space dimension is increased, and multiple ST problems are solved in the transformed space. At the end of the process, the predicted values are used to solve a linear system to obtain the original targets predictions.

Inspired by the related area of multi-label classification, Spyromitros-Xioufis et al. [14] proposed two techniques: SST (Stacked Single Target), also called MTRS (Multi-Target Regressor Stacking), and ERC (Ensemble of Regressor Chains).

The SST method consists of separately training ST models and using their outputs as additional prediction features. Thus, considering a dataset composed by $X = \{x_1, x_2, ..., x_n\}$ input features and $Y = \{y_1, y_2, ..., y_m\}$ target variables, SST uses the $Y^{'} = \{y_1^{'}, y_2^{'}, ..., y_m^{'}\}$ ST predictions as new features, forming a new training dataset $X^{'} = \{x_1, x_2, ..., x_n, y_1^{'}, y_2^{'}, ..., y_m^{'}\}$. The transformed input is utilized along Y values to train another regressors' layer, inducing new ST models, whose outputs are the final predictions. New income instances are first subjected to the first predictors' layer to obtain targets approximations and compose an augmented testing set, subjected to the second level of predictors. Although using ST estimations, the inter-target relationships are modelled and explored, consequently increasing the task's description capability and the prediction performance.

The ERC method consists of using a set of randomly chosen target chains to build ST models, following the generated sequence. For each chain, at first, an ST model is induced using the first output variable of the sequence. New models are trained following the chain order. Each new regressor uses an extended input dataset formed by the combination of the original input variables and the previous models' predictions. The described training process repeats until the end of the chain sequence. After training all models, new income instances are subjected to the set of chains. The final prediction for a target y is the average of the y predicted values over all chains. Since the output variables predictions come from the composition of values in different chains positions, multiple levels of combinations and interdependence among targets are investigated. In the original formulation, ERC explores all possible targets permutations if their number is less than 10, otherwise exactly ten random combinations are selected.

Although more than one predictor is used to represent the multiple targets problem, leading to decrease the model interpretation facility and increasing the computational training cost, this type of modelling offers several advantages. The possibility of using any base learner, even a hybrid set, could lead to better predictive performance and particular task's characteristics exploration. Besides that, adaptation techniques improve the solution's modularity and conceptual simplicity, having obtained significantly better accuracy than state-of-the-art methods [16, 14].

2.2 Deep Regressor Stacking - DRS

Our proposed technique applies the MTRS idea of using targets approximations as additional predicting features in a naive deep learning method. It is based on the hypothesis that the interaction among targets that happens in deeper layers could outperform the predictions obtained by none or only one prediction layers as input (ST or MTRS, respectively).

In this sense, Figure 1 presents the concept of Deep Regressor Stacking (DRS) multi-target regression. In ST method, the dataset's original attributes A are used to compute the prediction of the N targets ($T^1_{1...N}$). In its turn, MTRS predicts the targets using as input A and $T^1_{1...N}$, which means that the output predictions of $T^2_{1...N}$ are dependent, simultaneously, on the dataset attributes and the targets predictions of layer 1. Following the same logic, the DRS method will originate the prediction of the $(j+1)$-th layer using as input the attributes A and all the predictions from the j previous layers ($T^{1...j}_{1...N}$).

MTRS is a particular case of DRS, for the maximum of prediction layers used as input, j, equal to 1. By definition, ST uses no prediction layer as input, as already mentioned.

The Algorithm 1 demonstrates how to compute a price prediction based on DRS regression. The parameters of

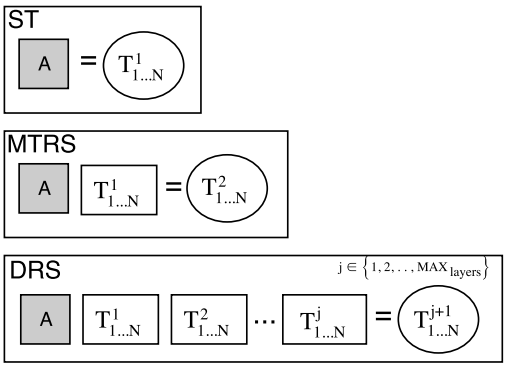

Figure 1: Comparison of ST, MTRS and DRS approaches.

the *price* function are the modelling dataset ($Data_{mod}$), the number of targets N, and the number of desired layers (λ). To be reasonable to refer to DRS, λ should be any natural greater than or equal to 3.

In the beginning of the algorithm, the dataset is split in two subsets: one for training ($Data_{tr}$) and the other for validation ($Data_{val}$).

$Targets$ is a vector representing the outputs, initially set with the N targets.

For N times the following procedure is adopted:

- DRS models are obtained with the training set, performed for λ layers. With the models, a prediction set P is resulted laying hand of D_{val}.
- In possession of P, the RMSE (Root Mean Square Error) between the target predictions and the output values are determined. The target with minimum RMSE value (t_x) and its corresponding layer (l) are recognized.
- The next step is training the dataset with a DRS structure again till the layer l. The prediction is calculated for the target t_x, and incorporated to the input set. In this training phase, if the length of $Targets$ is smaller than N, besides using the 1 to j previous layers' predictions as attributes, the predictions of the λ layer of the targets that were already combined in the input are also used as feature.
- Afterwards, the target that presented the smallest error is removed from the $Targets$ set and the model is saved to the *price* set, which will contain models.

Once the prediction of all targets were combined in the input (*i.e.*, $Targets$ is empty), the algorithm is finished.

Due to the stacking process, the dimensionality of this method increases significantly as λ and the number of outputs of the dataset increases, demanding a considerable processing time to obtain the final model.

Algorithm 1 Price prediction algorithm

function $price(Data_{\text{mod}}, N, \lambda)$
  $price \leftarrow \{\}$
  $\{Data_{\text{tr}}, Data_{\text{val}}\} \leftarrow split(Data_{\text{mod}})$
  $Targets \leftarrow \{t_1, t_2, ..., t_N\}$
  repeat
    $train \leftarrow \text{DRS}(Data_{\text{tr}}, \lambda)$
    $P \leftarrow predict(train, Data_{val_{input}}, \lambda)$
    $\{l, t_x\} \leftarrow \text{MIN}_{\text{RMSE}}(P, Data_{val_{output}})$
    $model \leftarrow \text{DRS}(Data_{\text{mod}}, l)$
    $T_x^\lambda \leftarrow predict(model, Data_{\text{mod}}, t_x)$
    $Data_{\text{mod}} \leftarrow \{Data_{\text{mod}}, T_x^\lambda\}$
    $Targets \leftarrow Targets - t_x$
    $price \leftarrow \{price, model\}$
  until $Targets = \{\}$
return *price*

3. EXPERIMENTAL SETUP

This Section describes the datasets used to compare the performance of the 4 different techniques, the base regression algorithms, and the software libraries employed.

3.1 Dataset

Two benchmark datasets of multi-target regression were explored in this work: ATP1D e ATP7D [1]. A summary of their attributes can be consulted in Table 1.

Name	Observations	Features	Targets
ATP1D	337	411	6
ATP7D	296	411	6

Table 1: Name, number of observations, features and outputs of ATP1D e ATP7D datasets.

ATP stands for Air Ticket Prices, and both have 6 target variables that represent flight preferences: any airline with any number of stops, any airline non-stop only, Delta Airlines, Continental Airlines, Airtrain Airlines and United Airlines. The main difference between ATP1D and ATP7D is that the first represent the target price in the next day; The last, the minimum price observed over the next 7 days. Among the input variables are present the number of days between the observation date and the departure date, the searching day of the week, the minimum price, mean price, and number of quotes from all airlines and from each airline quoting more than 50% of the observation days for non-stop, one-stop, and two-stop flights, for the current day, previous day, and two previous days [14].

These datasets were collected from a search website between February 22 and June 10, 2011 for 7 different origin–destination pairs (including major cities in different parts of the United States and some international destinations). The web spider used to extract the information is representative since it used the same information a costumer would have for acquiring the data [10].

With the goal of motivating the application of MT solutions to address the airline tickets prices predictions, we used two methods of statistical correlation assessment among targets variables of the analysed datasets: the correlation coefficients of Pearson and Spearman [3].

The Pearson coefficient measures linear relationships of continuous variables. A relationship among two outputs is linear when a change in one target is associated with a proportional alteration in the other.

The Spearman coefficient measures monotonic relationships

[1]The datasets can be downloaded from http://mulan.sourceforge.net/datasets-mtr.html

among continuous or ordinal variables. In a monotonic relationship, the targets should change together, but not necessarily at a constant rate.

Both metrics are equal to 1 when there is a perfect relationship among two variables (-1 if a perfect reverse relation is observed). When the coefficients are near to 0, there is no evidence of correlation among the observed variables. Comparing a target with itself will always generate a correlation coefficient equal to 1, for both methods.

Figure 2 shows the results of performed correlation tests for ATP1D dataset. Observing the coefficients results, it is possible to perceive high levels of linear and monotonic dependency among target variables in most of the cases, which is an indication of a MT problem.

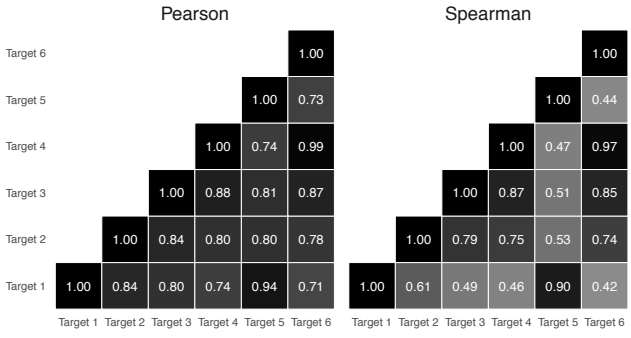

Figure 2: Pearson and Spearman correlation coefficients among ATP1D targets variables.

The same analysis was performed for the ATP7D dataset, whose correlation results are presented in Figure 3. For this dataset, it is possible to observe a decrease in both correlation coefficients when comparing with ATP1D results, which is a clue that the targets outputs are less correlated or there are levels of non-linear relationships among the output values.

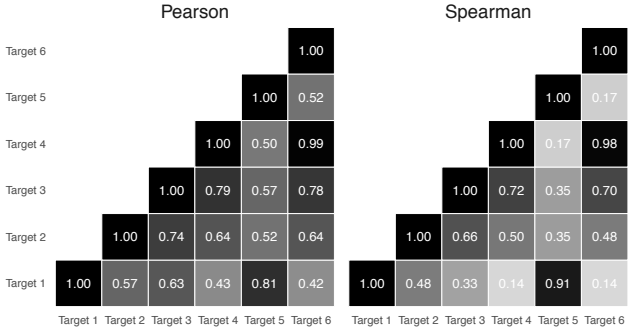

Figure 3: Pearson and Spearman correlation coefficients among ATP7D targets variables.

3.2 Parameters and Regression Algorithms

For the computation of DRS, the parameter λ should be pre-determined. For the tests of this work, the number of desired layers was set to 10. We wanted to use a number that was big enough to explore the deep dependencies and not too big to avoid a long time of computation.

Two regression algorithms from Machine Learning were used in the experiments: Support Vector Machine (SVM) and Random Forest (RF). Their wide use and different theoretical foundation motivated our choice.

All regression algorithms used in this work were implemented in R programming language, version 3.3.0, and used with their standard parameters settings. The packages *e1071* and *randomForest* were used for SVM and RF, respectively.

3.2.1 Support Vector Machine

The Support Vector Machine (SVM) is a classification and regression method belonging to the general category of kernel based methods. Its approach is based on maximizing the separation margin between classes, or minimizing the prediction regression error among training samples. Through kernel space transformation, this technique has the flexibility to model varied data sources [2], increasing the input dimensionality of data to a space where the separability is also increased.

3.2.2 Random Forest

Random Forest algorithm consists in independently growing decision trees based on different subsets of training data, formed by random sampling with replacement (Bagging). Each tree uses a subset of features randomly chosen. These procedures increases allow to explore different aspects of data, increasing the generalization capacity. The RF predictions are formed by taking the average result over all trees in the Forest [5].

3.3 Performance Metrics

To evaluate the models trained during the experiments three different performance metrics were used: Coefficient of Determination (R^2), average Relative Root-Mean-Square Error (aRRMSE), and Relative Performance (RP). Besides that, the multi-target techniques were performed using 10-fold cross-validation.

The RRMSE (Relative Root Mean Square Error) is calculated using the Root Mean Square Error (RMSE) of the predictions for a target divided by the RMSE of the average value of this output. This last acts as a baseline in the metric and allows the measurement of the improvement over a shallow predictor. This metric is very useful to compare non-homogeneous targets distributions and has been used in several multi-target works [4, 14]. The aRRMSE is defined as the average of the d targets RRMSE.

$$aRRMSE = \frac{1}{d}\sum_{i=1}^{d}\sqrt{\frac{\sum_{l=1}^{N_{test}}(y_i^l - \hat{y}_i^l)^2}{\sum_{l=1}^{N_{test}}(y_i^l - \overline{y}_i)^2}} \quad (1)$$

The Coefficient of Determination (R^2) explains the amount of the total variation associated with the use of an independent variable. Its values range from 0 to 1. The closer R^2 is to one, the greater is the quantity of the total variation in the output which is explained by the independent variables in the regression model [6].

The Relative Performance compares (RP) the aRRMSE of a single-target model with the aRRMSE of the other MT methods M (in our case, with MTRS, ERC and DRS), for d datasets. Thus, it can measure if there was an increase (RP greater than 1) or a decrease (RP lower than 1) relatively to the single-target results [14].

$$RP_d(M) = \frac{aRRMSE(ST)}{aRRMSE(M)} \quad (2)$$

The aRRMSE allows the comparison of possible technique superiority through the application of the Friedman's statistical test with significance level at $\alpha = 0.05$. The null hypothesis states that the performances of all compared multi-target techniques are equivalent regarding the averaged RRMSE per dataset. When the null hypothesis is discarded, the Nemenyi post hoc test could be applied, stating that the performance of two different models are significantly different whether the corresponding average ranks differ by at least a Critical Difference (CD) value. When multiple models are compared, a Critical Difference (CD) diagram could be used to represent the comparisons, as previously proposed in [7].

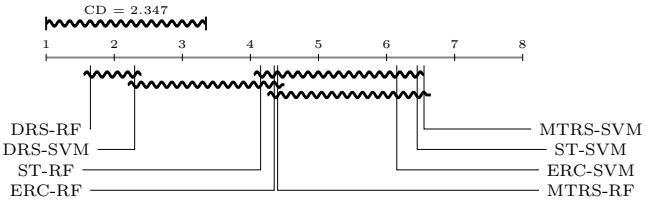

Figure 4: Comparison of the aRRMSE values per dataset for each CV fold configuration, according to the Nemenyi test. Groups of methods that are not significantly different (at $\alpha = 0.05$) are connected.

4. RESULTS AND DISCUSSION

After running the price prediction algorithm for the datasets ATP1D and ATP7D, and also ST, MTRS and ERC, statistical metrics were applied to the results.

The relative performance of the single-target method in relation to each multi-target method was registered in Table 2.

Table 2: Relative performance for ATP1D and ATP7D of ST in relation to MTRS, ERC and DRS methods.

Dataset	Regressor	**MTRS**	**ERC**	**DRS**
ATP1D	RF	1.0383	0.9549	1.9543
	SVM	1.2835	1.0256	1.8245
ATP7D	RF	0.9280	1.0700	1.6649
	SVM	1.0680	1.0548	1.7207

In ten out of the twelve values, the RP value was greater than 1. In other words, multi-target techniques outperformed single-target in 83,3% occurrences.

DRS had the best results among all combinations of methods, datasets and regressors. For particular cases, the equivalent DRS aRRMSE was reduced to almost the half of ST aRRMSE.

To propitiate a better comprehension of the results, the average coefficient of determination for the four methods were determined, as Table 3 shows.

Table 3: Average coefficient of determination for ATP1D and ATP7D using ST, MTRS, ERC and DRS methods.

Dataset	Regressor	**ST**	**MTRS**	**ERC**	**DRS**
ATP1D	RF	0.8535	0.8478	0.8436	0.9464
	SVM	0.7996	0.7960	0.8081	0.9315
ATP7D	RF	0.7701	0.7756	0.7735	0.8612
	SVM	0.6430	0.6335	0.6634	0.8826

The differences among ST, MTRS and ERC were subtle, with differences in the order of 10^{-2}. In contrast, the mean R^2 for DRS for the two datasets and regression algorithms were higher than the others in the order of 10^{-1}, which is a relevant difference since the possible R^2 value is in an interval of span 1.

For ATP1D, the overall R^2 was higher than for ATP7D. It was expected since the correlation among targets for the first (Figure 2) already indicated that.

The Nemenyi test was performed to verify if the discussed differences were statistically significant. According to Figure 4, with a significance level of 5%, the performance of DRS using as regressor RF presented no difference in relation to DRS with Support Vector Machine. DRS with RF was the the first in the rank, meaning that it had the best outcomes in relation to the others. In its turn, the critical distance of DRS with SVM showed that this method is comparable to ST-RF, ERC-RF and MTRS-RF.

On the account of what was evaluated, both DRS-RF and DRS-SVM outperformed ERC-SVM, ST-SVM, MTRS-SVM. This fact shows that RF was the best regression algorithm to interpret these datasets, and only DRS was able to obtain superior performance for SVM.

Apart from the presented performance advantages, DRS method had a particular drawback: the training phase requested plenty of time, even though using 10 layers. However, after the final model is complete, the application has a linear complexity. Additionally, in an ordinary system, the model is supposed to be created just one time (with possible re-training due to changes in the input information or modification in the data behaviour).

Figure 5 exemplifies how RMSE value has the possibility to decrease with the stacking of multiple layers, provided by DRS. For this, we used the prediction P recorded during training for the target LBL+ALLminp0+fut_001 (ATP1D) in an specific cross-validation fold, with random forest.

In the single-target prediction, the RMSE value was slightly bellow 0.042. In the second predictive layer, the RMSE dropped to around 0.019. In the third output layer, this value was even lower, bellow 0.01. This value continued decreasing in layers 4, 5 and 6. In the seventh layer the RMSE of this target increased, followed by a decrement in layer 8, an increment in layer 9, and again a decrement in the tenth layer, where the RMSE dropped to bellow 0.005, the lowest value among all layers.

The layers in which the growth in the amplitude of RMSE occurs are not necessarily the same. To exemplify this, the RMSE behavior in another fold for the same target, dataset and regressor was represented in Figure 6.

In this fold, the layer 8 interrupts the decreasing monotonicity, instead of the layer 7, as verified for the previous case.

It is questionable whether stopping the training in the fifth or in the tenth layer, for instance, would imply in extreme differences in the final results since their RMSE differences are not so significant. Thus, depending on the required accuracy of a problem, the choice of an optimal λ would be crucial to have the fastest model computation without af-

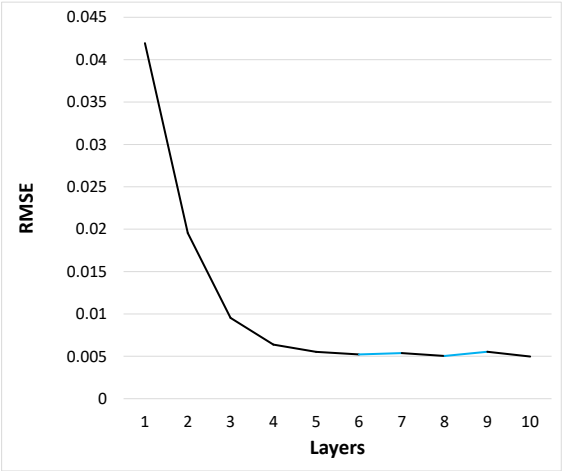

Figure 5: RMSE behavior for target LBL+ALLminp0+fut_001, from ATP1D, during training with random forest. Blue segments represent that the RMSE value of a higher layer was greater than the RMSE value of its immediate previous layer.

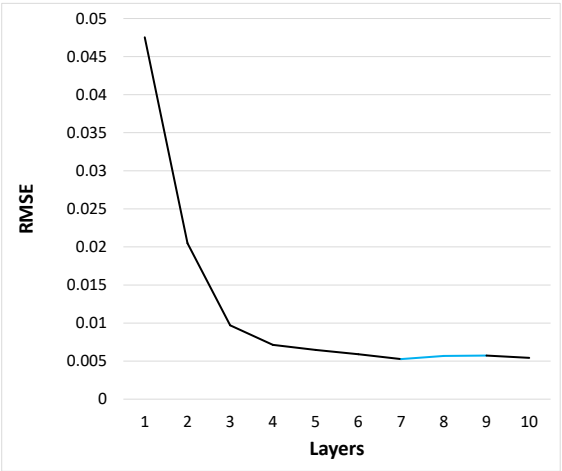

Figure 6: RMSE behavior for target LBL+ALLminp0+fut_001, from ATP1D, during training with random forest, in another CV fold. Blue segments represent that the RMSE value of a higher layer was greater than the RMSE value of its immediate previous layer.

fecting the quality of the prediction.

5. CONCLUSIONS

In this paper, we described a novel method (DRS) for improving the prediction of air ticket prices. Our original contribution towards Multi-Target prediction overperformed the state-of-art methods in two real-life datasets with two different learn-based algorithms.

The next step would be the implementation of a system with the DRS as the kernel. The output screen would essentially display to the user 3 columns, one for the current price, another for the for the next-day price and a final for the minimum price over the following 7 days. The system would also be capable of extracting some features automatically, for example, the day of the week and the ticket prices of all airlines present in the dataset output, similar to the current systems, although more accurate.

The authors affirm that DRS could be also used for predicting other Multi-Target scenarios. Besides the choice of a λ, another suggestion of future work is testing DRS with different problems that evolve price prediction.

Referências

[1] International Air Transport Association. Annual Review 2016. Technical report, Dublin, 2016.

[2] Asa Ben-Hur and Jason Weston. A user's guide to support vector machines. *Data Mining Techniques for the Life Sciences*, pages 223–239, 2010.

[3] Douglas G. Bonett and Thomas A. Wright. Sample size requirements for estimating pearson, kendall and spearman correlations. *Psychometrika*, 65(1):23–28, 2000.

[4] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233, 2015.

[5] Leo Breiman. Random forests. *Machine learning*, 45 (1):5–32, 2001.

[6] JA Cornell and RD Berger. Factors that influence the value of the coefficient of determination in simple linear and nonlinear regression models. *Phytopathology*, 77(1): 63–70, 1987.

[7] Janez Demšar. Statistical comparisons of classifiers over multiple data ss. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[8] Oren Etzioni, Rattapoom Tuchinda, Craig a. Knoblock, and Alexander Yates. To buy or not to buy: Mining airfare data to minimize ticket purchase price. *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining*, (August):119–128, 2003.

[9] William Groves and Maria Gini. A regression model for predicting optimal purchase timing for airline tickets. *Technical Report*, 2011.

[10] William Groves and Maria Gini. On Optimizing Airline Ticket Purchase Timing. *ACM Trans. Intell. Syst. Technol. 7, 1, Article 3*, 7(1):1–28, 2015.

[11] Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. Ensembles of multi-objective decision trees. In *European Conference on Machine Learning*, pages 624–631. Springer, 2007.

[12] Dragi Kocev, Sašo Džeroski, Matt D White, Graeme R Newell, and Peter Griffioen. Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling*, 220(8):1159–1168, 2009.

[13] Guangcan Liu, Zhouchen Lin, and Yong Yu. Multi-output regression on the output manifold. *Pattern Recognition*, 42(11):2737–2743, 2009.

[14] Eleftherios Spyromitros-Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis Vlahavas. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98, 2016.

[15] Joanna Stavins. Price Discrimination in the Airline Market: The Effect of Market Concentration. *The Review of Economics and Statistics*, 83(1):200–202, 2001.

[16] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Aikaterini Vrekou, and Ioannis Vlahavas. Multi-target regression via random linear target combinations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–240. Springer, 2014.

[17] Till Wohlfarth, Stéphan Clémençon, François Roueff, and Xavier Casellato. A Data-Mining Approach to Travel Price Forecasting. *ICMLA*, 2011.

[18] Tao Xiong, Yukun Bao, and Zhongyi Hu. Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. *Knowledge-Based Systems*, 55:87–100, 2014.

[19] Shuo Xu, Xin An, Xiaodong Qiao, Lijun Zhu, and Lin Li. Multi-output least-squares support vector regression machines. *Pattern Recognition Lett.*, 34(9):1078–1084, 2013.

[20] Wei Zhang, Xianhui Liu, Yi Ding, and Deming Shi. Multi-output LS-SVR machine in extended feature space. *CIMSA 2012 - IEEE Int. Conf. Comput. Intell. Meas. Syst. Appl. Proc.*, pages 130–144, 2012.