Interpreting financial time series with SHAP values

Karim El Mokhtari
Data Science Laboratory, Ryerson University
Toronto, Ontario, Canada
elmkarim@ryerson.ca

Ben Peachey Higdon
Data Science Laboratory, Ryerson University
Toronto, Ontario, Canada
bpeachey@ryerson.ca

Ayşe Başar
Data Science Laboratory, Ryerson University
Toronto, Ontario, Canada
ayse.bener@ryerson.ca

ABSTRACT

We apply SHAP values to explain how non-linear models predict commentaries on financial time series data. We show how SHAP values are used to assess the usefulness of additional datasets and how they significantly improve the accuracy of tested models. Our industrial partner uses non-linear models to predict commentaries by learning from financial experts reports. Even though a good accuracy has been reached, management wants to demystify the prediction process and needs to demonstrate whether a new and hardly accessible dataset can be useful in prediction. We create an explanation model based on SHAP values to reveal the predominant features and to demonstrate the contribution of the new dataset. This explanation model is also applied to reveal what specific features trigger each class of commentary. We show that new dataset does not improve the learning and that financial experts often rely on specific months to write their commentaries. We also show how SHAP values can be useful in improving the prediction accuracy as they naturally cluster datapoints according to feature importance.

KEYWORDS

SHAP values, times series, model interpretation

ACM Reference Format:
Karim El mokhtari, Ben Peachey Higdon, and Ayşe Başar. 2019. Interpreting financial time series with SHAP values. In *Proceedings of CASCON 2019.* Toronto, ON, Canada, 7 pages.

1 INTRODUCTION

Creating forecasts and comparing them to actual results is vital to every company in the market. Our industrial partner is a consumer goods company that sells hundreds of products to customers worldwide. These customers are primarily supermarket chains and retailers. To monitor the performance of every product, monthly reports are generated to show sales volumes by product and customer. These reports are then aggregated by brands of products, and total sales volumes are compared with the company forecast to show the discrepancy between actual and forecast. Experts analyze the discrepancies and write short commentaries to describe the reason of every alarming deviation and the main customers behind it. To validate their findings, they often turn to managers in various departments such as customer service or warehouse for inquiries on a specific product or brand.

This process relies heavily on the expert's knowledge of the business areas, but it depends on data living in the different silos of the company as well. That is why our partner's management expressed the need to create a model to learn from existing commentaries and reveal any particular pattern triggering the generation of commentaries.

In our previous paper [1] our main focus was on learning and predicting commentaries using NLP tools and recurrent neural networks. However, due to the insufficiency of commentaries from different categories and the diversity in the writing style and semantics between experts, the model was only able to predict very basic commentaries.

As many commentaries are related to sales performance, we discussed with our partner to evaluate the Point of Sales (POS) data as a candidate dataset. It is extracted from the database used by customers to daily manage their retail stores. It contains the time and place of all retail transactions. Access to POS data requires a special agreement between our partner and the customer and poses multiple security concerns. That is why, in this work, we use POS data from only one important customer to assess its usefulness.

In addition, our partner is interested in interpreting how the different models used predict a commentary, and what particular patterns in the monthly results trigger commentary generation.

To achieve both objectives, we have reformulated the problem by categorizing commentaries into multiples classes. We evaluate several machine learning models in commentary prediction. Then, we use explanation models to understand how every model computes the output. There are many methods to interpret a model prediction, but we found that SHAP (SHapley Additive exPlanation) values proposed by Lundberg and Lee [7] are more consistent as we will explain in Section 2.

This paper is organized as follows. In the next section, we present briefly the methods used in explaining models and the motivation behind choosing SHAP values. In section 3, we explain the methodology followed in assessing the relevance of the POS dataset and in explaining how every model computes its output. In section 4, we describe the experimental part and show the results. The conclusion summarizes the findings and discusses future works.

CASCON 19, November 2019, Toronto, Ontario, Canada

2 MODEL PREDICTIONS INTERPRETATION

It is easy to interpret and understand the prediction of linear models. However, in the non-linear case, the model itself cannot be used in interpretation as the prediction process is often considered as a black box. Therefore, we define an explanation model as an interpretable approximation of the complex model as illustrated in Figure 1.

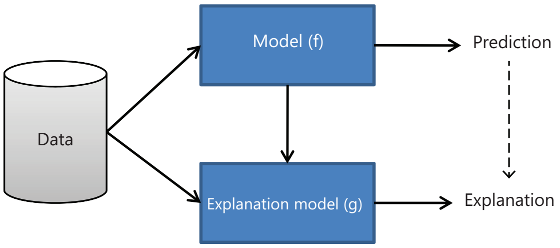

Figure 1: How an explanation model is used in predicting interpretation

2.1 Additive feature attribution methods

As proposed in [2], local methods are designed to explain the prediction model f by the explanation model g. Model g uses simplified inputs x' mapped to the original inputs via a function $x = h_x(x')$ specific to every original input x. Local methods try to approximate $f(h_x(z'))$ by $g(z')$ when $z' \approx x'$. In additive feature attribution methods, the explanation model is a linear combination of M binary variables z' that represent a feature being observed when $z' = 1$ or unknown otherwise:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{1}$$

M is the number of simplified input features. Coefficients ϕ_i are real numbers that show the effect of each feature. Summing up all the effects results in an approximation of the prediction $f(x)$ that we want to explain.

Several methods were proposed to match equation 1. The LIME method [2] finds ϕ_i by minimizing an objective function ξ described by equation 2 based on a squared loss L and a local kernel $\pi_{x'}$ to weight the simplified inputs. It uses a penalized linear regression in the minimization problem.

$$\xi = \arg\min_{g \in \mathcal{G}} L(f, g, \pi_{x'}) + \Omega(g) \tag{2}$$

DeepLIFT was proposed in [3, 4] as another additive feature attribution method that applies a recursive algorithm to explain deep neural networks prediction. It uses a "summation-to-delta" property to match equation 1.

The above methods rely on equations from cooperative game theory to compute prediction explanations using Shapley regression values [6], Shapley sampling values [4] and Quantitative input influence [5]. Shapley regression values consists in retraining the model f on every feature subset S from $F \setminus \{i\}$, the set of all features F excluding i. To calculate the effect of feature i, two models are trained, one including the feature $f_{S\cup\{i\}}$ and the second f_S withholding it. The difference between the output of both models $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)$ computed for a specific input x_S shows the effect of feature i on the prediction. As the order of withholding feature i can affect the prediction in case feature i depends on other features in subset S, the model is retrained for all possible subsets $S \subseteq F \setminus \{i\}$. Equation 3 calculates Shapley values as a weighted average of all possible differences.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \tag{3}$$

Shapley sampling values [4] reduce the high computational load of retraining the model on all combinations of S in equation 3. The method approximates the effect of excluding a feature by integrating over samples from the training set.

Lundberg and Lee [7] numbered three properties in the class of additive feature attribution methods: local accuracy, missingness and consistency. They demonstrated that methods not based on Shapley values violate at least one of these properties. They proposed a unified measure of feature importance called SHAP values that we explain in the next section.

2.2 SHAP values

It is demonstrated in [7] that equation 4 corresponds to the only explanation model that satisfies local accuracy, missingness and consistency.

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)] \tag{4}$$

$|z'|$ is the number of non-zero entries in z', and $z' \subseteq x'$ stands for all z' vectors where the non-zero entries are a subset of the non-zero entries in x'.

SHAP values are defined as a solution to this equation where f_x is a conditional expectation function of the original model: $f_x(z') = f(h_x(z')) = E[f(z)|z_S]$ with S the set of non-zero indexes in z' as illustrated in Figure 2. However, exact computation of SHAP values is challenging. That is why authors in [7] proposed several approximations assuming model linearity and feature independence. A version of SHAP values adapted to Tree ensembles was proposed recently in [8, 9].

In this paper, we use KernelSHAP that is based on LIME [2]. The choice of the objective function parameters in equation 2 is not made heuristically, which violates the consistency property, instead it uses the following parameters proposed in [7] to recover the Shapley values:

$$
\begin{aligned}
\Omega(g) &= 0 \\
\pi_{x'} &= \frac{(M-1)}{(M\ choose\ |z'|)|z'|(M-|z'|)} \\
L(f,g,\pi_{x'}) &= \sum_{z' \in Z} [f(h_x^{-1}(z')) - g(z')]^2 \pi_{x'}(z')
\end{aligned}
$$

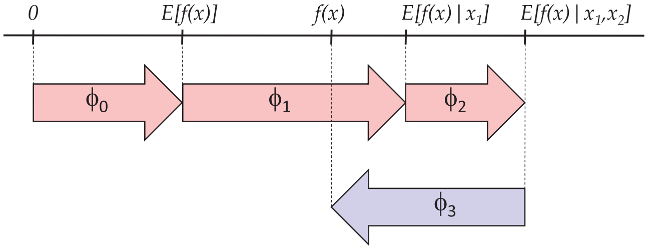

Figure 2: How SHAP values explain the output as a sum of effects ϕ_i of each feature i for a single prediction

SHAP values were also applied in supervised clustering experiments. It is shown in [8, 9] that transforming original attributes into SHAP values results in a unit-less multidimensional space where only feature importance is expressed. Such transformation mitigates the challenging problem in unsupervised clustering that is determining feature weightings. In this paper, we do not use supervised clustering, but we run another classification algorithm on the computed SHAP values to assess whether using this new harmonized set of attributes would increase the performance of the new classifier in comparison with the original classifier operating on raw attributes.

3 METHODOLOGY

In this study, we are interested in predicting commentary generation on financial data using binary classifiers. We use SHAP values theory to interpret the output of the prediction model. Additionally, we want to show the impact of the new POS dataset on the prediction accuracy. As mentioned above, this dataset requires access to the customer sales database, a special agreement and security considerations between our partner and its customers. SHAP values are applied to explain the output computation for every model.

We use the discrepancy dataset denoted VAR extracted from the company's ERP (Enterprise Resource Planning). It shows the monthly variance from January 2016 to present by brand and customer (see Table 1). The experts commentaries are provided monthly in the COM dataset shown on Table 2. The commentaries explain the performance of one or many customers, with a brief description of the root cause of the discrepancy observed for each one of them. Our goal is to create a model that learns to generate a commentary from the VAR dataset.

In our previous paper [1], the focus was on generating plain English commentaries with RNN structures. In this paper, we categorize commentaries into different classes then we experiment with several models. We use k-Nearest Neighbours (kNN) and Support Vector Machines (SVM), two well described algorithms often used for classification problems. We use two ensemble methods: Random Forest and XGBoost, a gradient tree boosting algorithm [10]. Generally, we can expect ensemble methods to outperform single learners [11]. Lastly, we implement a Long Short-Term Memory network (LSTM) [12], a recurrent neural network that excels at capturing patterns in sequential data as is the case with our monthly data.

Table 1: The VAR dataset describes discrepancies observed by brand for a chosen customer from January 2016 to present in Millions of CAD

CUSTOMER 1

Brand	Jan. 2016	Feb. 2016	...	June 2019
Brand 1	+0.10	-0.01	...	+0.05
Brand 2	-0.08	-0.05	...	+0.12
...	...	...	...	...

CUSTOMER 2

Brand	Jan. 2016	Feb. 2016	...	June 2019
Brand 1	-0.03	+0.11	...	-0.03
Brand 2	+0.14	-0.08	...	+0.06
...	...	...	...	...

Table 2: The COM dataset is an aggregation of VAR by brand and month, it shows discrepancies in Millions of CAD for each brand along with the financial expert commentaries

January 2016

Brand	Variance	Commentary
Brand 1	+0.50	Variance driven by CUSTOMER 10 and CUSTOMER 25, caused by over delivery
Brand 2	-0.71	CUSTOMER 15: Promotion in brand did less than expected
...	...	...

June 2019

Brand	Variance	Commentary
Brand 1	-0.39	CUSTOMER 12: High inventory
Brand 2	+0.63	CUSTOMER 10 and CUSTOMER 25: declining faster than seen in the market
...	...	...

We assume that a commentary depends mainly on the variance recorded in the current month along with data coming from the twelve months preceding its generation. Consequently, we process the VAR dataset as follows. For each brand b and month m, and for each commentary $C_m^{(j,b)}$ generated for customer j, we compute an input vector $V_m^{(j,b)}$ of 13 items described as follows:

$$V_m^{(j,b)} = \left[v_{m-12}^{(j,b)}, v_{m-11}^{(j,b)}, \ldots, v_{m-1}^{(j,b)}, v_m^{(j,b)}\right] \tag{5}$$

where $v_{m-k}^{(j,b)}$ is the value of the discrepancy recorded for brand b and customer j on month $m-k$.

The POS dataset is also aggregated by month and brand and for each commentary $C_m^{(j,b)}$ generated for customer j, we construct a vector $P_m^{(j,b)}$ as follows:

$$P_m^{(j,b)} = \left[p_{m-12}^{(j,b)}, p_{m-11}^{(j,b)}, \ldots, p_{m-1}^{(j,b)}, p_m^{(j,b)}\right] \tag{6}$$

where $p_{m-k}^{(j,b)}$ is the total value of sales recorded for all products under brand b and sold by customer j during month $m-k$.

We train the models first with the VAR dataset alone, then with both VAR and POS datasets. Each model performs a binary classification where 1 indicates that a commentary needs to be generated. We measure the performance of every model using the F1-score defined as the harmonic mean of precision and recall.

We explain every model using SHAP values to evaluate feature importance and POS dataset relevance. To assess the contribution of features on every class, we apply the same models in a multiclass classification framework. In this case, every model is trained to predict one of six classes of commentaries; Promo, POS, SP&D, Phasing, Other and NoComm. The "Other" category includes rare commentaries that are not related to any of the main categories. The NoComm corresponds to cases where no comment is generated. This category is required for models to identify patterns that do not trigger any expert reaction. However, it is frequent that no comment is generated in normal situations. This results in an imbalanced dataset where most data points have no related comment. We randomly oversample the less frequent categories as a simple solution to overcome the effects of the class imbalance [13].

Finally, we apply the idea behind supervised clustering in [8, 9] by computing SHAP values for the whole VAR dataset. Instead of using clustering, we train the same group of binary classifiers using SHAP values as input this time. We assume that this SHAP value transformation improves the learning process as it removes units from the dataset and emphasizes the most relevant features. We compare the performance of two groups of binary classifiers, the first uses the VAR dataset as input and the second the SHAP values computed from this dataset. We note here that both groups are assessed with the same test set whose real labels are unknown for both of them.

4 EXPERIMENT

We choose different classes of classifiers using different working algorithms: kNN, SVM, XGBoost, Random Forest and LSTM. We run the following experiments:

- Binary classification for commentary prediction with VAR only and both VAR and POS datasets
- Multiclass classification for commentary class prediction
- Binary classification using SHAP values as input

Table 3: F1-score of binary classifiers using VAR alone and VAR+POS datasets

	kNN	RF	SVM	XGB	LSTM
F1-score VAR	0.72	0.69	0.71	0.67	0.66
F1-score VAR+POS	0.53	0.65	0.61	0.64	0.51

After normalizing both VAR and POS datasets and splitting data into a training set and a test set with a 80/20 ratio, we define a pipeline for every model as the follows.

(1) Balance the dataset by randomly oversampling the less frequent commentaries
(2) Run Grid-search to compute the optimal classifier parameters
(3) Train the model with KFold cross-validation with 100 runs of random oversampling and compute the F1-score in every run
(4) Compute the mean of F1-scores for all runs. This is the overall F1-score reported for the model

4.1 Binary classification

The F1-scores reported on Table 3 show that the prediction performance degrades when the POS dataset is included. kNN and SVM performs best with VAR while RF and XGB perform equally well with VAR+POS.

We compute SHAP values for every classifier using the KernelSHAP method described in Section 2.2 available in the SHAP python package published by Lundberg [14] . Figure 3 illustrates the contribution of every feature with the SVM classifier. Features are sorted by their importance. Every dot corresponds to a datapoint of VAR. Red dots have higher commentary generation probability. We notice that an increase in the discrepancy of the same month of last year (M-12) and the following month (M-11) increases the probability of a commentary generation.

We calculate a weighted average of the mean of SHAP values computed for every classifier. We apply a model vote scheme using F1-scores as weights to combine all SHAP values from every classifier and for each feature. Figures 4 and 5 show that the discrepancy of M-12 is the most important feature in generating a commentary across all models. Comparison of feature importance between VAR and POS datasets in Figure 5 shows clearly that, in the overall, models rely more on the first dataset than on the second. This confirms the decrease in performance when POS dataset is included.

4.2 Multiclass classification

In the multiclass classification case, models predict the class of the generated commentary. The performance metric that we use here is F1-score macro denoted $\mathcal{F}_1$ which is the mean of the F1-scores

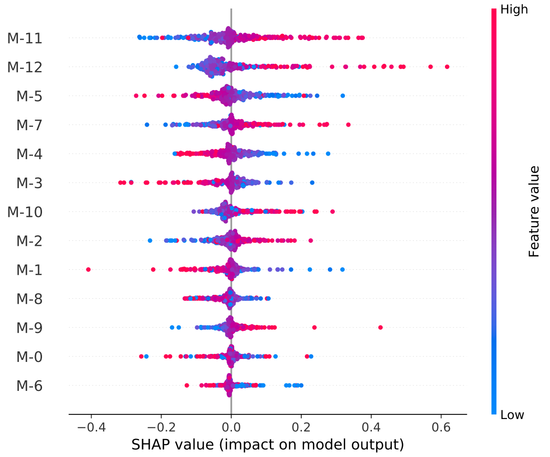

Figure 3: SHAP values plot explaining the SVM classifier prediction. $M - k$ stands for k months preceding the current month

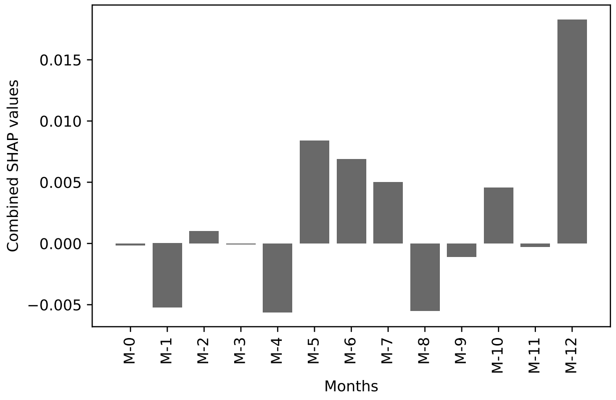

Figure 4: weighted average vote of SHAP values with VAR dataset

by class defined below.

$$\mathcal{F}_1 = \frac{1}{|C|} \sum_{c \in C} F1_c \qquad (7)$$

where C is the set of classes to predict and $F1_c$ the F1-score computed for a class $c \in C$. This score is impacted whenever the F1-score of a class drops.

Table 4 shows the performance of multiclass classifiers with both VAR and VAR+POS datasets. The performance remains the same or drops when POS dataset is included. This experiment shows again that models do not learn much from this dataset. Figure 6 shows the SHAP values computed for each class with the kNN model that performed best on the VAR dataset. We observe the predominance of M-12 for the frequent classes Promo, POS, SP&D and NoComm, while months M-7 and M-9 were predominant in "Other" and "Phasing" classes respectively. These results shed light on the way the financial expert handles data. Models confirm that last year performance for any brand is one of the main drivers in the commentary generation process, while some specific months are of more importance for less frequent commentary classes.

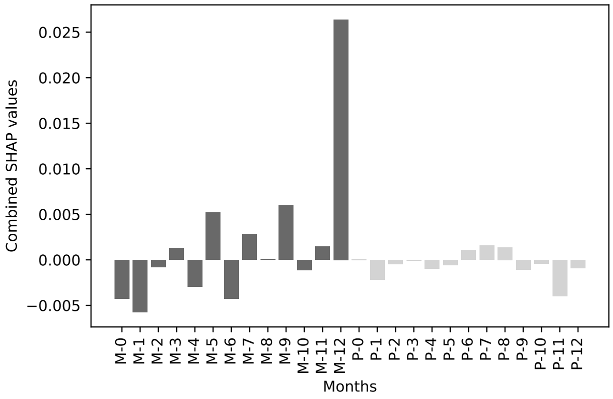

Figure 5: weighted average vote of SHAP values with VAR+POS datasets

Table 4: F1-score macro $\mathcal{F}_1$ of multiclass classifiers using VAR alone and VAR+POS datasets

	kNN	RF	SVM	XGB	LSTM
$\mathcal{F}_1$ VAR	0.34	0.20	0.15	0.19	0.16
$\mathcal{F}_1$ VAR+POS	0.15	0.18	0.15	0.19	0.16

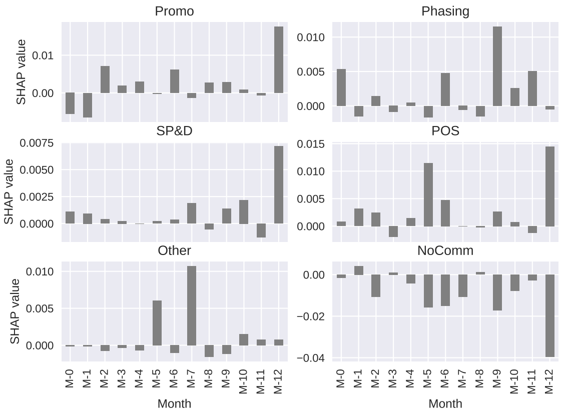

Figure 6: SHAP values by class computed for the kNN classifier with VAR dataset

4.3 Binary classification on SHAP values

In this last experiment, we use the SHAP values computed for the kNN binary classifier that performed best in section 4.1. We apply these values as inputs to the same group of binary classifiers to measure the impact of harmonizing features through SHAP values explained in [8, 9]. In this experiment, the same test set is used on the classifiers with VAR dataset and to the ones using the SHAP values dataset.

Figure 7 shows a significant improvement in the performance of all classifiers after applying SHAP values transformation. This can be explained by the fact that SHAP values operate a natural clustering of datapoints and thus allow the model to learn better than on the original dataset.

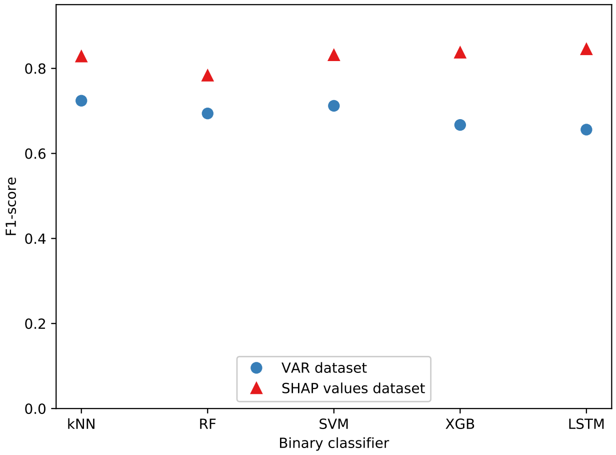

Figure 7: Comparison of two groups of binary classifiers: one using the VAR dataset and one using SHAP values computed from this dataset. The two groups are tested with the same test set

5 THREATS TO VALIDITY

It is important before concluding this work to point out some main threat to validity that we describe below.

- The dataset is imbalanced due to the presence of an important number of datapoints without commentaries. To overcome this threat, we applied random oversampling to minor classes. We also chose F1-score as a metric to penalize the model performance if it did poorly on either precision or recall.
- In multiclass classification, we applied F1-score macro averaging which penalizes the overall score if the F1-score drops for any class. This allows to give more weights to classes that might be less represented in the test set
- To reduce the effect of any specific model in computing the SHAP values of each attribute, we used a vote scheme where SHAP values computed from different models were combined by a weighted average
- In the last experiment on the impact of the SHAP values transformation, we used the same test set as in the experiment with original values so that the test set labels remain unknown to both groups of classifiers.

6 CONCLUSION

The aim of this work is to apply SHAP values to explain the prediction of models dealing with financial time series. The main dataset provided by our industrial partner is the monthly discrepancy (VAR) between forecast and actual results for by brand of products, along with expert commentaries on the performance of each brand. Our partner wants to assess the usefulness of the POS dataset and needs explanations on specific data patterns that lead to the generation of a commentary.

Several binary classifiers were tested, kNN and SVM performed best. Including the POS dataset did not improve models learning. This was confirmed by interpreting models output with SHAP values that showed the prediction relied more on the actual vs forecast discrepancy than on POS data. Additionally, through the multiclass classification and SHAP values we proved that the performance of a brand in the same month of last year can be considered as the main trigger in generating most classes of commentaries.

Finally, we demonstrated that SHAP values used as a transformation of the original dataset apply a natural clustering that has a positive impact on the commentary prediction accuracy. Future works include considering inventory as a potential new candidate dataset and applying time series classification algorithms to increase accuracy.

ACKNOWLEDGMENTS

This work is supported by grants from MITACS and Smart Computing For Innovation (SOSCIP) consortium.

REFERENCES

[1] El Mokhtari K., Maidens J., Bener A. (2019) Predicting Commentaries on a Financial Report with Recurrent Neural Networks. In: Meurs MJ., Rudzicz F. (eds) Advances in Artificial Intelligence. Canadian AI 2019. Lecture Notes in Computer Science, vol 11489.

[2] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?: Explaining the predictions of any classifier". In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. 2016, pp. 1135-1144.

[3] Avanti Shrikumar et al. "Not Just a Black Box: Learning Important Features Through Propagating Activation Differences". In: arXiv preprint arXiv:1605.01713 (2016).

[4] Erik Strumbelj and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions". In: Knowledge and information systems 41.3 (2014), pp. 647-665.

[5] Anupam Datta, Shayak Sen, and Yair Zick. "'Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems"'. In: Security and Privacy (SP), 2016 IEEE Symposium on. IEEE. 2016, pp. 598-617.

[6] Stan Lipovetsky and Michael Conklin. "Analysis of regression in game theory approach". In: Applied Stochastic Models in Business and Industry 17.4 (2001), pp. 319-330.

[7] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems 30. Curran Associates, Inc., 4768-4777. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[8] Lundberg, S.M., Erion, G.G., Lee, S.I.: Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888 (2018)

[9] Lundberg, S.M. and Lee, S.I.: Consistent feature attribution for tree ensembles. arXiv preprint arXiv:1706.06060, 2017

[10] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM. 2016

[11] Zhou, Zhi-Hua. Ensemble methods: foundations and algorithms. Chapman and Hall/CRC, 2012.

[12] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation. 9, 1735-1780 (1997).

[13] Haibo He and Edwardo A. Garcia. "Learning from imbalanced data." In: IEEE Transactions on knowledge and data engineering 21.9 (2009): pp. 1263-1284.

[14] Lundberg SHAP github: https://github.com/slundberg/shap