Shapley values for future and ensamble models

Davide Franzoso
davide.franzoso.2@studenti.unipd.it

Alessandro Benetti
alessandro.benetti.1@studenti.unipd.it

Abstract

This project is aimed to tackle the problem of explainability for black-boxes machine learning models, especially for ensemble model and for the features within a dataset.

In order to accomplish this task we are going to compute a metric for each sub-model that compose an ensemble (explainability for ensemble) or feature (explainability for feature) called shapley value, that is a game theory concept. We will see that this metric can be used for ensemble building, ensemble description or for feature analysis.

1. Introduction

The advent of black boxes machine learning models raised fundamental questions about their understandability. This is a very serious and still the center of many studies (especially for deep learning models), in fact, one would know why a specific model gives in output a particular result and which are the factors that it considered. We frame this question as one of the valutation of features in a dataset and models in an ensemble.

Feature valutation The behavior of machine learning models in taking out their decisions is treated as black-box hindering the interpretability of their decisions. In this project we use shapley values to explain the behavior of a classifier model in discriminating if a person survived or not when the Titanic sunk. In order to do that we used a specific open-source library for python called Shap (https://github.com/slundberg/shap Lundberg and Lee (2017)). Moreover, we based our work on the paper proposed by Haneen Alsuradi (2020)

Ensamble valutation We introduce ensemble games, a class of transferable utility cooperative games. Each classifier in the ensemble receives the data point, and they output, for each data point a probability distribution over the classes, the final decision will be give by taking in consideration every single distribution. Using these values an oracle quantifies the worth of each model in the ensemble (shapley values). This value are important in order to make some decision such as ensemble building. In order to compute the shapley values for an ensemble games we based our work on Rozemberczki and Sarkar (2021) and, from a practical point of view, we used the library shapley

2. Related works

The Shapley value is a solution to the problem of distributing the gains among players in a transferable utility cooperative game . It is widely known for its desirable axiomatic properties such as efficiency and linearity. However, exact computation of Shapley value takes factorial time, making it intractable in games with a large number of players. General and game specific approximation techniques have been proposed. Shapley values can be approximated using a Monte Marlo MC sampling of the permutations of players. A more tractable approximation is using a multilinear extension (MLE) of the Shapley value. The only approximation technique tailored to weighted voting games is the expected marginal contributions method (EMC) which estimates the Shapley values based on contributions to varying size coalitions. One of the methods implemented (shapley values for ensemble) is built on EMC.

Shapley value has previously been used in machine learning for measuring feature importance. In the feature selection setting the features are seen as players that cooperate to achieve high goodness of fit. Another machine learning domain for applying the Shapley value was the pruning of neural networks. It is argued in that pruning neurons is analogous to feature selection on hidden layer features. Finally, there has been increasing interest in the equitable valuation of data points with game theoretic tools. In such settings the estimated Shapley values are used to gauge the influence of individual points on a supervised model. In this sense, value of data and how individuals should be compensated has been intensely discussed by economists and policy makers.

Shapley value was proposed in a classic paper in game theory and has been widely influential in economics. It has been applied to analyze and model diverse problems including voting, resource allocation and bargaining. Parallel works have studied Shapley value in the context of data

valuation focusing on approximation methods and applications in a data market. In linear regression, Cook's Distance measures the effect of deleting one point on the regression model. Leverage and influence are related notions that measures how perturbing each point affects the model parameters and model predictions on other data. These methods, however, do not satisfy any equitability conditions, and also have been shown to have robustness issues.

3. Shapley values

The shapley value is a solution concept in cooperative game theory. Basically for each possible cooperative game among the player the shapley value assign a unique distribution of a total surplus generated by the coalition of the players. So, the shapley values are very important in the context of cooperative game in order to understand which are the players that contribute more to the coalition than others or may posses different bargaining power. How can this concept be used in the context of machine learning? There are many scenario in machine learning that can be view as a cooperative game, for example ensamble models and features. In the ensemble model scenario we have a set of binary classifiers (which form the ensemble) that play a cooperative voting game to assign a binary label to a data point by utilizing the features of the data point.
A similar scenario can be describe for a features in a dataset: Each features can be seen as a player in a cooperative game where each player tries to predict the right class. In this sense we can compute the shapley value for each feature (given a data point) in order to evaluate the most influential feature.

4. Methods implemented

This project is based on the work proposed by Rozemberczki and Sarkar (2021) and Haneen Alsuradi (2020) in order to compute the shapley values of an ensemble model and for a set of features respectively.

4.1. Shapley values for ensembe

We used the library shapley in order to compute the shapley values for an ensemble. shapley is the official implementation of the work proposed by Rozemberczki and Sarkar (2021). Before computing the shapley values themself we have to perform the following point:

- Define a binary ensemble machine learning model M with m sub-models m_i.
- Train M on a dataset $\{(x_1, y_1), ..., (x_n, y_n)\}$ where $x \in \mathbb{R}^d$ and $y \in \{0, 1\}$
- Compute for each sub-model in M, given a data sample (x_i, y_i), its weight in the ensemble. Now we give the definition for the individual model weight:
  An individual weight of the vote for m_i, in a sub-ensemble $S \subseteq M$, for a data point (x_i, y_i) is defined as:

$$w_{m_i} = \begin{cases} \text{P(y = 1} \mid m_i, x_i)/m & \text{if y}_i = 1 \\ \text{P(y = 0} \mid m_i, x_i)/m & \text{otherwise} \end{cases} .$$

  Note that the weight of m_i depends on the size of the larger ensemble M
- After computing the weight for each individual model and for each data sample we have a matrix A of size mxn where $m =$ number of samples in the dataset and $n =$ number of classifiers (players in the game) in the ensemble.
- Once obtained the weight matrix for the ensemble model we are ready to compute the shapley values, a matrix B that has the same dimension of the weight matrix A.

The matrix B is computed using the *Troupe* algorithm, method proposed by Rozemberczki and Sarkar (2021). Since the computation of the exact shapley values is a NP problem, *troupe* algorithm compute just an approximation of them. There are many ways to approximate the shapley values, but, since ensemble game is a variant of voting game, we can use *the expected marginal contribution(EMC)* approximation. Basically the algorithm iterates over the individual model weight and it calculates the expected contribution of the model to the ensemble. We can see this value as the probability of the model to become the marginal voter of the game.

4.2. Shapley values for features

As already said we have computed shapley values for both ensemble and features. For the first task we have used shapley, a library that requires the steps described before in order to reach our goal. For the second task instead we used *Shap* Lundberg and Lee (2017) an higher level library used primarily for computing shapley values for features. Shap uses a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic shapley values from game theory and their related extensions. This is a high level library so it is required less steps in order to compute the desired metrics:

- Just like for the ensemble case we have to define a machine learning model and training it using some data X
- After that, we have to use some data X_t (sampled from the same distribution) in order to build a background distribution for the framework.

- Now we are ready to compute the shapley values using X and X_t using the function *shap.explainer()* and *explainer()*

So the core idea behind Shap is to use fair allocation results from cooperative game theory to allocate credit for a model output $f(x)$ among its input feature.

5. Experiments

We have performed experiments for both ensemble task and features task. We will see that shapley values is a useful metrics for ensemble building and feature extraction. For both task we have choosen as dataset the titanic dataset (https://www.kaggle.com/c/titanic) where, using 891 sample composed of 11 features (for example: age, place of embarkation, class, fare price ecc..), it associates a boolean variable that describe if a specific passenger survived or not.

5.1. Experiments for ensamble

In this section we are going to explain why the shapley values is a useful metrics when we have to deal with ensembles models. In fact, they are useful in order to perform ensemble building task and to understand the complexity of each sub model.

5.2. Ensemble building

With the term "ensemble building" we refer to the practice of selecting (based on a metric) a high performance subset M' of the original ensemble M such that M' outperforms M

Experimental settings We have performed the ensemble building operation by constructing M' in a forward fashion, starting from the original ensemble M. In such operation we have used the shapley values as metric to select this high performance subset. The dataset used, for both training and evaluation of the performances, is the "titanic dataset" and, as original ensemble M, we have decide to use the *random forest* ensemble with 200 trees. M is given using the default settings of *scikit-learn* (Pedregosa et al. (2011)). The procedure is composed of the following steps:

- We have trained the model using the 70% of the original dataset. Each sub-tree is trained by selecting a random subset of features, in order to guarantee the generalization and avoiding overfitting.
- For each sub-tree its corresponding weights have been computed.
- By selecting the 15% of the rest of the dataset we have retrieved the average shapley value H_i for the sub-tree t_i among all the samples.
- The trees $(t_1....t_n)$ have been ordered, in decreasing order, respect to their associated average shapley value. So, after this operation, we have obtained a list T where $H_i >= H_{i+1}$ where H_i is the shapley value associated to the sub-tree $t_i \in T$
- After that, we built $|T|$ ensembles in a forward fashion using the list of sub-trees T. At each iteration j the ensemble built is composed of: $(t_1....t_j)$ and we evaluate it by computing its accuracy. The accuracy is retrieved using the remained 15% of the dataset.

As we can see in Figure 1, the ensemble score tends to increase up to a certain number of sub-trees (80) and then it decreases. This is due to the fact that in the tail of the list T we have trees with low shapley values and so they haven't got importance in the classification process or, they can misclassify some samples.

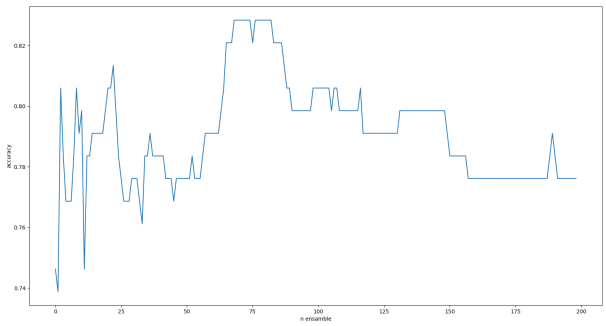

Figure 1. Ensemble building

5.3. Model complexity and influence

The shapley values are used as a metric to evaluate the importance of a specific model in a ensemble model. It would be interesting to understand if there is a correlation between the complexity of the sub-model (i.e the depth of a tree or the number of hidden neurons in a neural network) and its respective average shapley value among the samples. In order to perform this experiment we have decided to build a soft voting ensemble, composed of 500 neural networks. Each neural network has 1 hidden layer and a number of hidden neurons that is randomly draw from a list $N = \{2^2, 2^3, 2^4\}$ and it is trained using the standard settings of *scikit-learn* with a maximum of 500 epochs.

Experimental findings In general, there is a correlation between the complexity of the model and the average shapley value associated to it. This is due to the fact that a more complex model is going to have an higher weight w_{m_i} in the ensemble respect to the models with a lower complexity, and this result in a higher shapley values. So complex

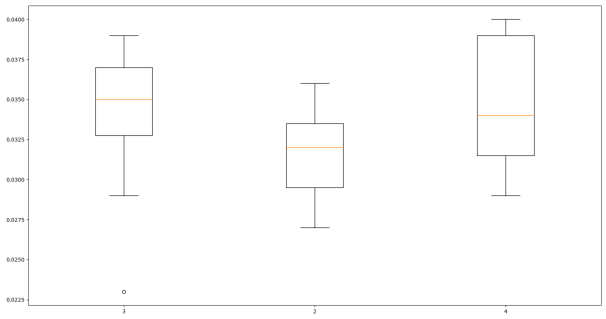

Figure 2. On the y-axis we have the average shapley values among the samples, on the x-axis the base 2 logarithm computed on the number of hidden neurons

models contribute to the correct and incorrect classification decisions at a disproportional rate. This result can be seen on Figure 2, where we plotted the boxplot of the average shapley values associate to neural networks with 3 different number of hidden neurons.

5.4. Experiments and discussion for features

In this section we are going to show how the shapley values, computed on the features of a dataset, can be a useful metric in order to understand the decision process of any machine learning model. We will present some graphs that show the importance of the feature (given a model) and how these graphs can be used in order to perform feature extraction. As for ensemble the experiments were performed using the titanic dataset.
This choice is given from the fact that our experiments can be supported from intuitive thoughts about the features (for example a female had an higher possibility to survive respect to a male). Respect to the the ensemble task, here we have performed the experiments using two models: a support vector machine and a random forest, in this way we can see how the features have different weights on the decision process of different models.

5.5. Feature level analysis

Since we have used two models we are going to analyze their results separately. For both models we have computed the same graphs with some little variations. We have analyzed the shapley values of features given a single sample x_i and in general.

Support vector machine Before starting the experiments we have decided to delete those feature that are not necessary for the classification (i.e understanding if a passenger survived or not), specifically we have deleted the name of the passenger, the number of the ticket and the number of the cabin. We have decided however to maintain a useless feature (PassengerId) to understand if we were able to trick one of the classifiers and if we were able to see the aftermath of this choice also on the computed shapley value.

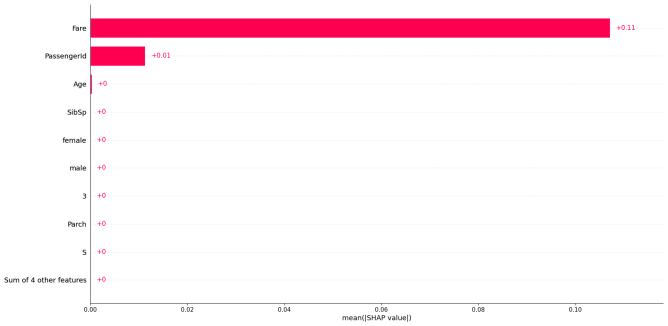

Figure 3. Bar plot that represents for each feature the average shapley value among all the samples.

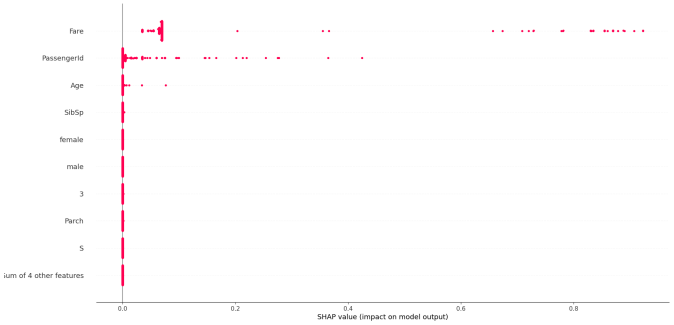

Figure 4. Beeswarm plot that represents for each feature the absolute average shapley value among all the samples

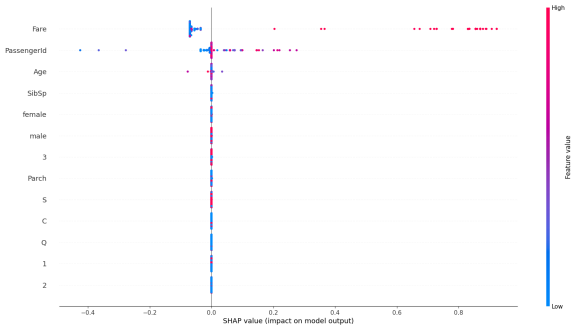

Figure 5. Beeswarm plot that represents for each feature the shapley value among all the samples

As we can see from Figure 3, Figure 4 and Figure 5 the PassengerId feature is the second feature that the model took in consideration in the decision process and it didn't take in consideration other important features like the age or the sex of the person. If we take a look to the Figure 5 it seems like there is a correlation between the greatness of the PassengerId and the probability to survive. This is clearly false and so this fact affect heavily the generalization of the model, in fact, we pass from an accuracy of ≈ 0.65 to an accuracy of ≈ 0.70 by excluding the PassengerId feature.

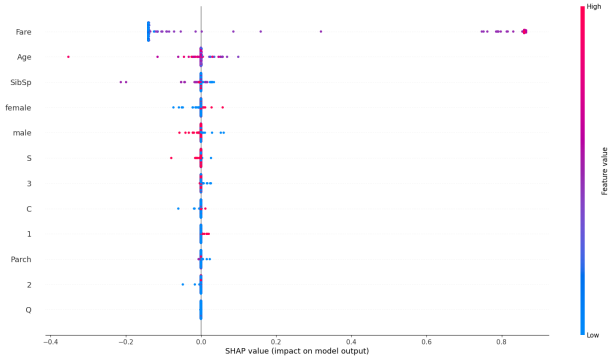

Figure 6. Beeswarm plot that represents for each feature the shapley values among all the samples (without PassengerId feature)

From figure 6 we can notice that the classifier, without the PassengerId feature, takes in consideration the fare paid from the passenger as major indicator to understand if she/he survived or not (as before) but also other factors like the age (lower value: higher possibility to survive) or the sex (the women had an higher chance to survive) .

Random forest ensemble The support vector machine model works pretty well but we have noticed that it didn't consider many important features, for example, if the passenger travelled in first or third class. So, instead of considering just a model we have considered a plurality of models using an ensemble, the random forest ensemble. In the next pages we are going to present some graphs that explain how different features weight on the ensamble model decision.
As we can see from Figure 7, Figure 8 and Figure 9 if we use an ensemble, there are a lot of more samples (dots in the graph) that have an equally distribution of the features' weights in the decision process of the model respect to the SVM. The fact that more features have more weight in the ensemble, can be translated in the fact that the ensemble takes in consideration more factors in order to perform the classification. This consideration helps for sure the generalization of the model, that results in an higher accuracy respect to the support vector machine: we pass from an accuracy of ≈ 0.70 (SVM) to an accuracy of ≈ 0.81 (ensemble).

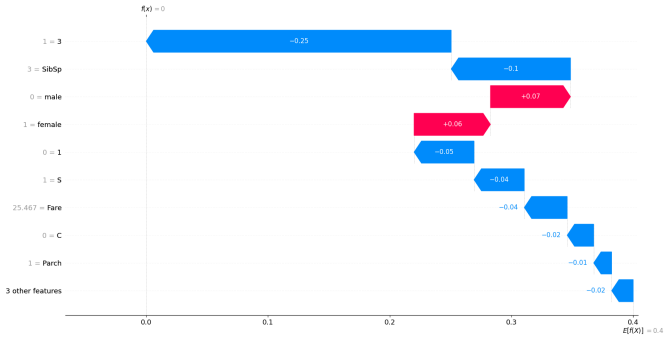

Figure 7. Waterfall graph that represents how different features "move" the prediction for a specific sample x_i from the expected value of all the prediction in the dataset $E[f(x)]$. In red we have those features that increase the probability to survive, in blue those features that decrease the chance to survive. For this specific sample, the fact that he/she travelled in third class (3=1) has decreased a lot the probability of surviving

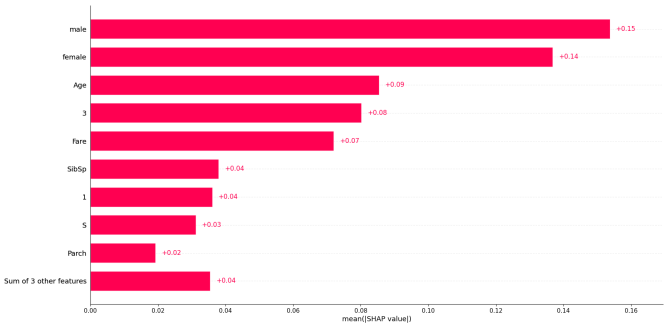

Figure 8. Bar plot that represents for each feature the absolute average shapley value among all the samples. As we can see from this plot the ensemble model takes in cosideration a lot more features respect to the SVM model.In average, the sex and the age of a person plays an important role to determine her/his surviveness, then the model takes in consideration in which class the person travelled, if she/he has siblings and so on.

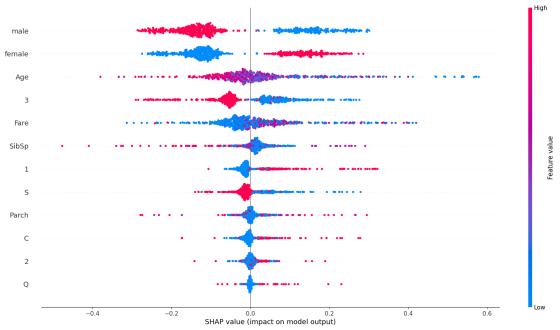

Figure 9. Beeswarm plot that represents for each feature the shapley values among all the samples. As Figure 8 suggest us the ensemble model takes in consideration a lot more features respect to SVM model. In this plot we can see if a feature increase or decrease the chance of surviveness respect to its value. For example the fact that a person is not a male (so the feature male has a low value because male = 0) increase the probability to survive. Notice that for the second feature, the feature female, it's the opposite.

6. Conclusion

As we have seen during this relation the computation of the shapley values are useful in many different ways: starting from the mere interpretability of a machine learning model to the enhancing of its performance (ensemble building, feature analysis). We have analized how the shapley values can be computed for an ensemble model (submodels as players) and for features (features as players), this reasoning can be extended also for the dataset itself. The idea is to compute the shapley values for each sample (data as players) in order to understand which kind of sample has more weight in the decision process. This extension is studied by Ghorbani and Zou (2019)

References

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning, 2019.

Mohamad Eid Haneen Alsuradi, Wanjoo Park. Explainable classification of eeg data for an active touch task using shapley values. 2020.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Benedek Rozemberczki and Rik Sarkar. The shapley value of classifiers in ensemble games. *CoRR*, abs/2101.02153, 2021. URL https://arxiv.org/abs/2101.02153.

shapley. Shapley. https://github.com/slundberg/shap.