IAA AAI

SECTION COLLOQUIUM 2019

THE MODERN ACTUARY - CHALLENGE • INFLUENCE • LEAD

ASTIN • IAAHS • IAALS • IACA • PBSS

ISOA

2019

Cape Town
South Africa

CTICC

www.colloquium2019.org.za

Hosted by

ACTUARIAL SOCIETY OF SOUTH AFRICA

A Standardized Machine Learning based approach to Conversion Rate Estimation

The aim of this presentation is to describe a **Standardized Machine Learning Based Approach** to the **Conversion Rate** estimation exploiting the most advanced techniques available in the Data Science Field, but <u>taking into account the possibility to deploy into production the optimal estimated model</u>

1. Combining a cross validation approach with an Automated Bayesian Approach, we obtain the “best” prediction from **five different models**
2. Using simulations, a weighted average of the five singular models was calculated, proving that all models are sub optimal (i.e. **Two Layer Ensemble Model – Type 1**)
3. Starting from the most significant features detected using the Shap Value and the five predictions of the models as new features, a second layer LightGBM model is trained (i.e. **Two Layer Ensemble Model – Type 2**)

Lorenzo Invernizzi
Generali Italia
lorenzo.invernizzi@generiali.com

Vittorio Magatti (presenter)
Willis Towers Watson
vittorio.magatti@willistowerswatson.com

Introduction, definitions and foundation of our research

The **Conversion Rate** is defined as the **ratio** between the **number** of the underwritten **insurance policies** and the number of the **quote requests**:

- **0 ("zero")** – even if some potential clients ask for a quote, they decide to buy another insurance proposal
- **1 –** for each quote request there is a **new** insurance policy

Both of the above cases can clearly be defined as extreme cases, but represent the **range** of this indicator

A good prediction of this ratio produces at least two main advantages for an Insurer:

1. ***Increase in Competitiveness***: this is especially important when the underwriting cycle shows a softening period
2. ***Effective price changes***: a Company could identify rate changes or dedicated discounts coherently with the estimated conversion and profitability calculated for each potential client, both needed to develop a pricing optimisation tool

The analysis were carried out in **Python**, using licenses open source libraries heavily used by practitioners and trusted by the Data Science community

Data and selected perimeter of our investigation

- The selected perimeter is the **Motor Third Party Liability** (MTPL) for private cars. The MTPL (all vehicles) in Italy represents the 41% of the Non-Life Gross Written Premium *(*)*
- Database is founded on a real aggregated data set representing an Italian benchmark market

- The **train** set is composed by 520,325 (≈ 80% of the data) quote requests. The average observed/historical conversion rate in this is 24.3%
- The **test** set is composed by 130,081 (≈ 20% of the data) quote requests. The average observed/historical conversion rate in this is 23.9%

- For each quote request **26 features** are considered: *premium range, age of the client, power-to-weight ratio, Bonus Malus, engine power, vehicle age, years of car ownership, vehicle age at the purchase date, occupation, guide style, age of patent qualification, housing density, horse powers, Italian region, number of non insured years, marital status, fuel type and education*
- In order to treat properly all variables, numerical features are encoded into ordered integer after creating bins based on their distribution, while we apply *One Hot Encoding* on the categorical variables

(*) ANIA, "Premi del lavoro diretto italiano 2017" - http://www.ania.it/export/sites/default/it/pubblicazioni/rapporti-annuali/Volumi-Premi-lavoro-diretto-italiano/2017/PREMI-2017-x-WEB.pdf

Machine Learning Models (1/5)

Below the models used to build the **First Layer of the Ensemble Model** are introduced, highlighting their main properties

Generalized Linear Model - GLM

The GLM represents the state of the art algorithm extensively used in the Insurance sector to predict the Conversion Rate. The Binomial family distribution is considered as the error distribution, in association with its canonical Logit link function. See Chapter 2 of [4] for a thorough discussion of the statistical model

$$y_i = x_i^{'}\beta + \varepsilon_i$$

Classification and Regression Tree – CART (*)

Let $\{Ai\}_i$ be a partition of the 26 dimensional space of the features, the CART is defined as a linear combination of indicator functions

$$CART(x) = \sum_i c_i \mathbf{1}\{x \in S_i\}$$

The model fits by minimizing a specified loss function and is able to **capture non-linear** and **complex relationships**. In contrast there is a **high risk of overfitting**. See Section 9.2 of [5] for more details.

(*) It is not treated as a singular Machine Learning model, but it is the base of the models reported in the next slide

Machine Learning Models (2/5)

Random Forest - RF

The Random Forest consists of an **average** of K CART models

$$\frac{\sum_k CART_k(x)}{K}$$

where each CART is estimated by means of **two random effects**: *Bootstrapping* (≈ 70%) and Feature Bagging (√26). Both hyperparameters, important in preventing overfitting, are subject to fine tuning. See [3] for a complete argumentation.

Gradient Boosting Machine - GBM

The Gradient Boosting Machine is defined as a **linear combination** of CART models

$$BOOST(x) = sigmoid\left(\sum_k \alpha_k CART_k(x)\right)$$

where the sigmoid function is used to map the combination of CARTs into [0;1].

Therefore the model tries to iteratively adjust the prediction by fitting a **gradient** on the mistakes made in previous iterations. See [1] for more details.

Machine Learning Models (3/5)

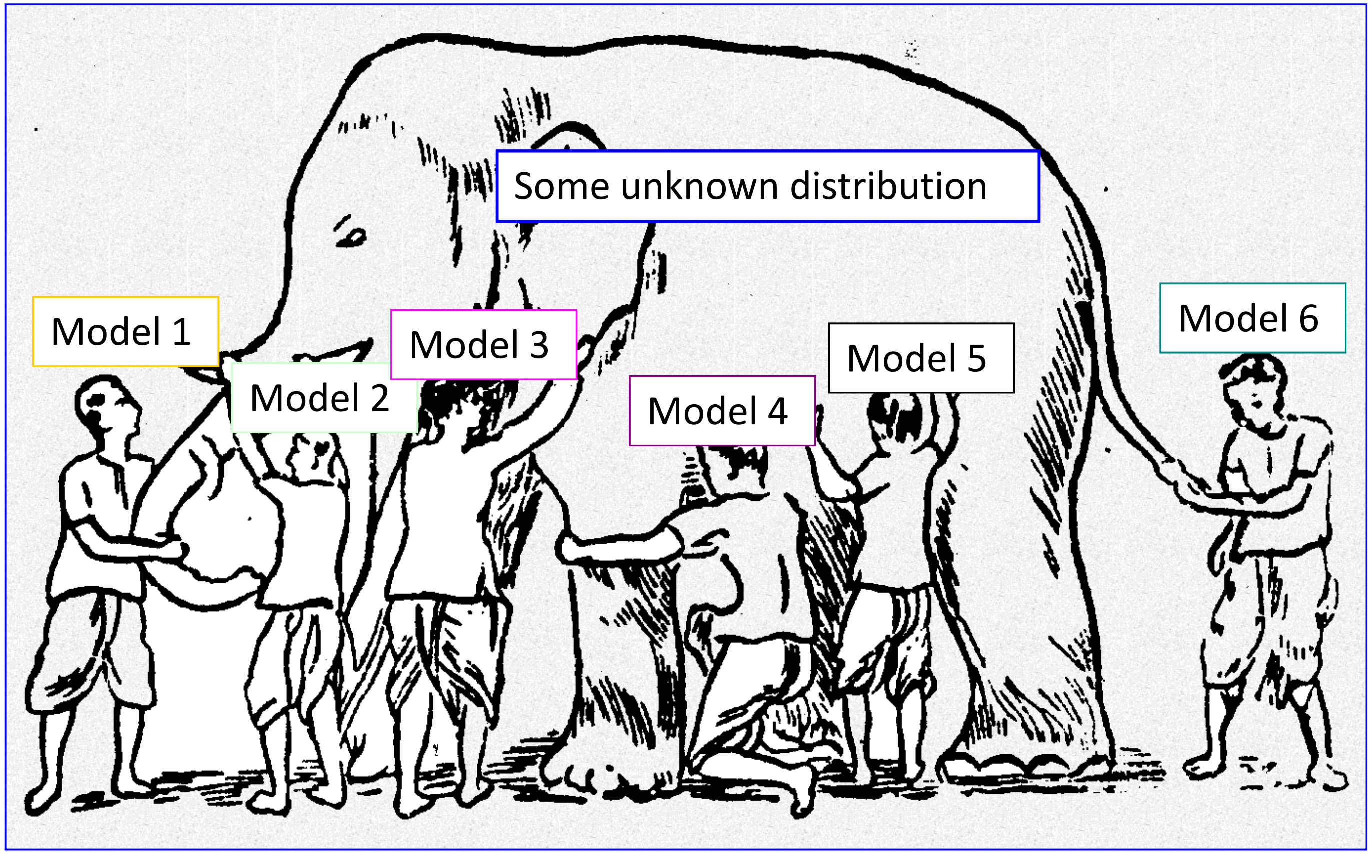
From the GBM to the Gradient Boosting Decision Trees (GBDT)

- CART models are estimated by minimizing a specified loss function
- There is no method that can find the best split while avoiding going through all features of all data points

Therefore, the various implementations of Gradient Boosting Decision Trees (GBDT) are methods of finding the approximate best split. We selected the most well known:

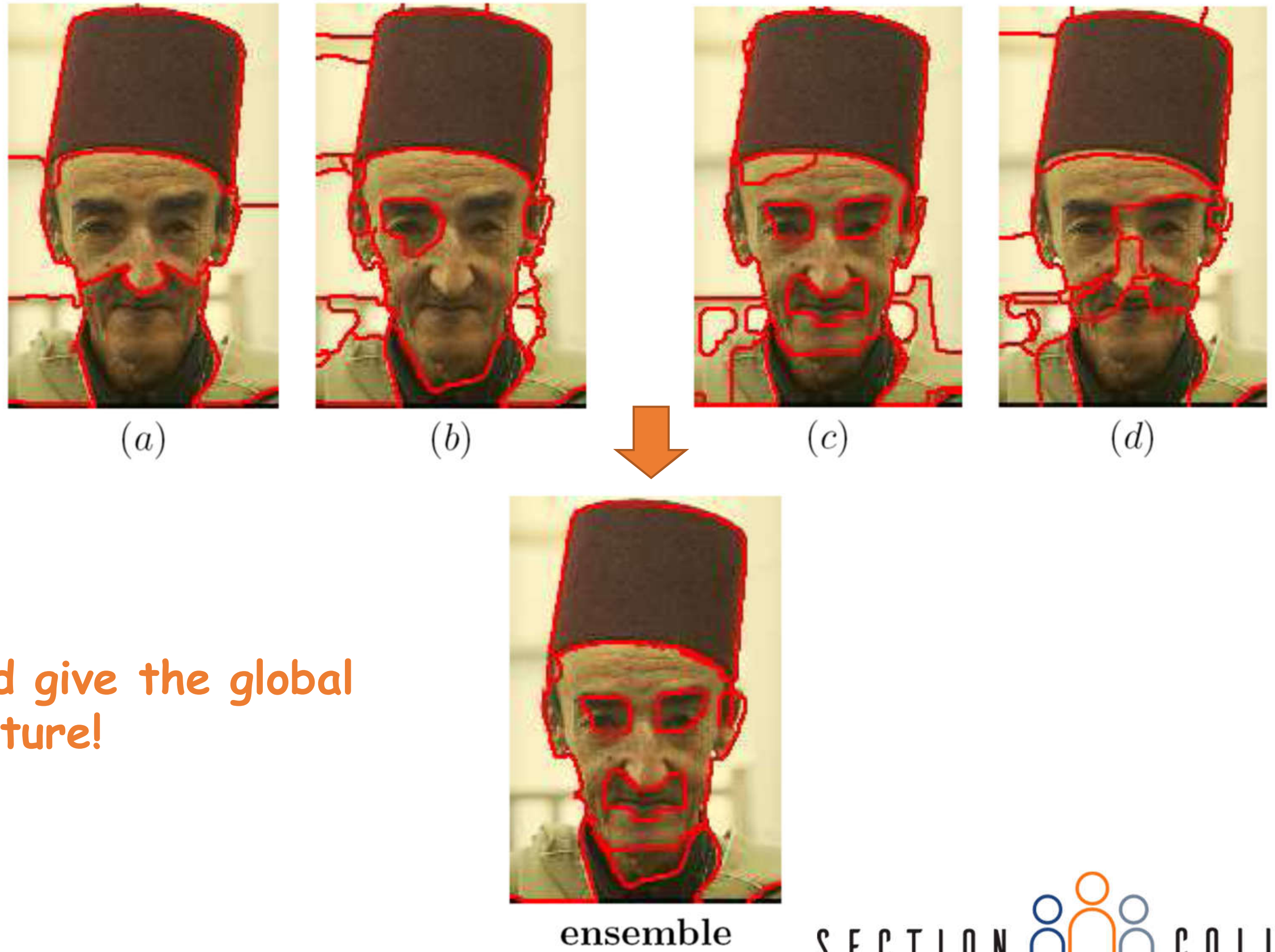
- ***XGBoost*** - it implements Histogram-Based methods to approximate the best split and ignores sparse inputs. See [8] for the complete algorithm
- ***LightGBM*** - it implements the same methods of XGBoost, plus a method called Gradient-based One-Side Sampling used to sample data based on their gradient. See [9] for a complete explanation
- ***CatBoost*** - it exploits Histogram-Based methods as the previous two implementations, but the main advantage is that it is able to automatically handle categorical features without any explicit pre-processing ([10])

Machine Learning Models (4/5)



Ensemble could give the global picture!

Machine Learning Models (5/5)

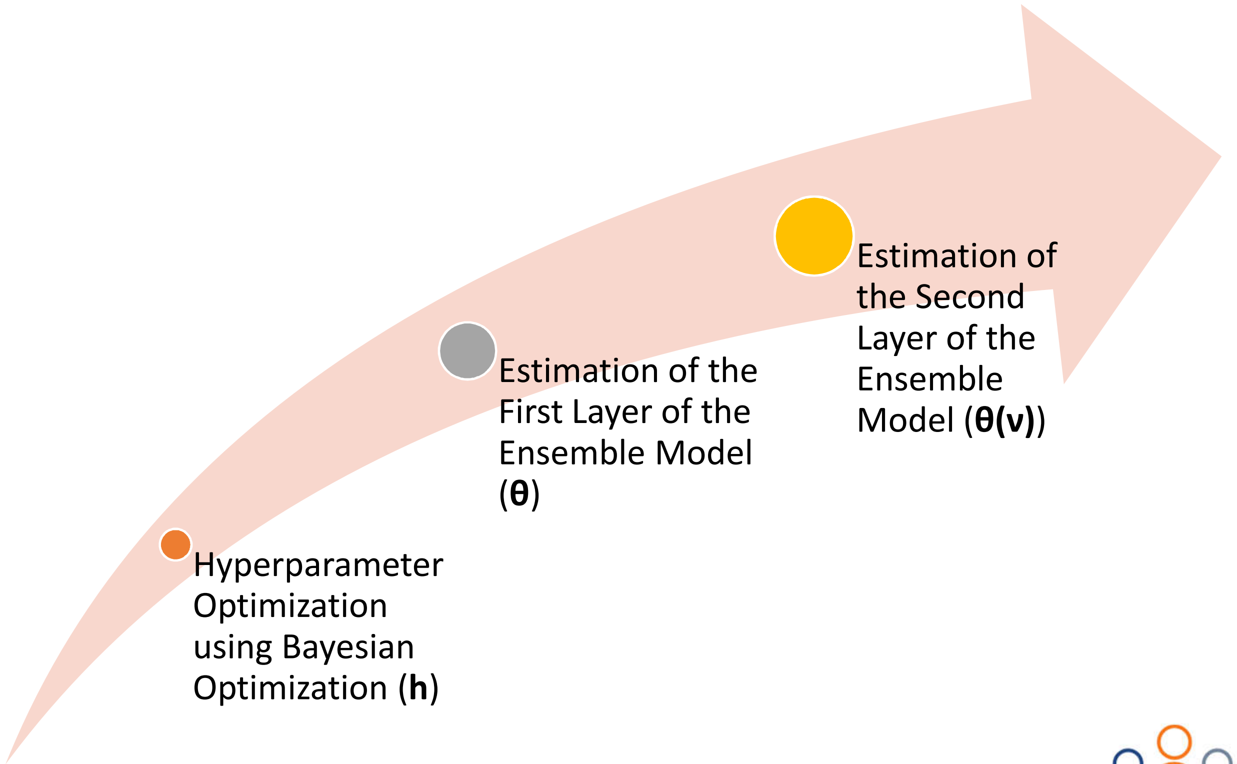


Ensemble could give the global picture!

SECTION COLLOQUIUM 2019

Methodology two calibrate a “Two Layer Ensemble Model”

As reported in the first slide, the **methodology** we present in order to calibrate a Two Layer Ensemble Model $\{Model_i(y, \theta_i, h_i)\}$ can be divided into **three** main **steps**

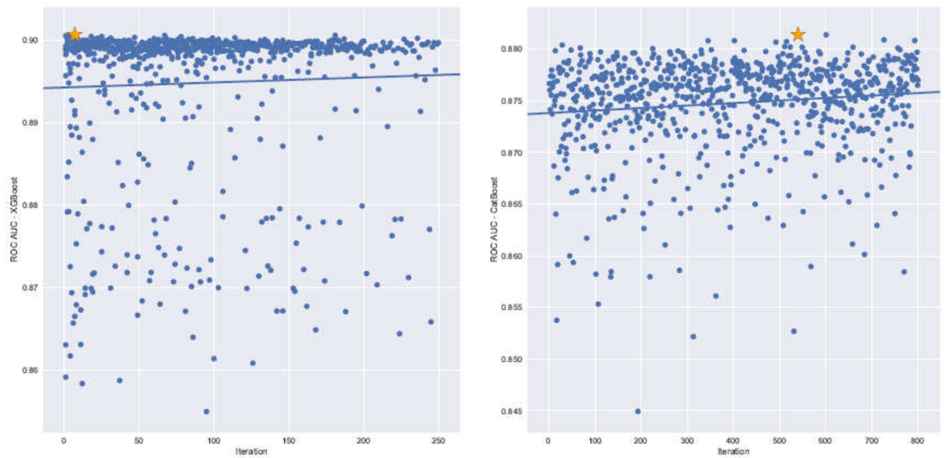


Hyperparameter Tuning using Bayesian Optimization

Find the hyperparameters that yield the lowest error on the validation set in the hope that these results generalize to the testing set

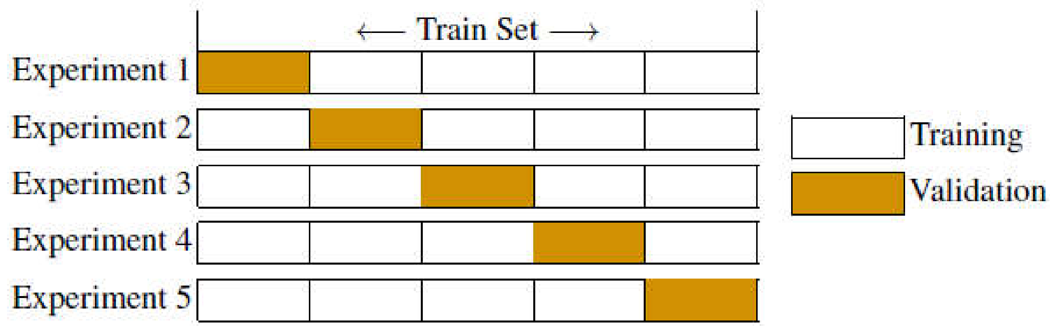
1. Grid (or Full) search
2. Random search
3. Bayesian Optimization (see [6] for more details)

- **Limit** expensive evaluations of the objective function by **choosing** the **next input** values based on **those** that have done **well in the past**, where the objective function is the Cross Validation Error of a Model using a set of hyperparameter
- Plots of ROC AUC score against the search iteration using Bayesian Optimization. As you can see, there is a **positive correlation between the number of iteration and the score**. Stars in the plots represent the highest value attained. For each ML Model, the number of maximum iterations carried out depends on the computational time. On average, **for each model one day of computations is needed to complete the search.**



First Ensamble Layer

The strategy adopted to create the first ensemble layer consists of dividing the training set into 5 folds as shows in the table below



At this point, in order to estimate the parameters **θ_i** for each of our ML model in $\{Model_i(y, \theta_i, \widehat{h_i})\}$, we do the following:

1. For Experiment = 1 to 5 do:
    - Fit $Model_i(y, \theta_i, \widehat{h_i})$ on the 4 folds (white or training) → $Model_{ij}(y, \widehat{\theta_{ij}}, \widehat{h_i})$
    - Use this estimated model to predict on the 5th fold (orange, or validation).
2. Combine the 5 disjointed set of predictions of the models $\{Model_i(y, \theta_{ij}, \widehat{h_i})\}_{j=1:5}$ into one complete out-of-fold prediction of the training set → **v_i**

Second Ensamble Layer

!

The estimation of the Second Ensemble Layer consists of consider the out-of-fold predictions $\{v_i\}_i$ generated by each i-th model as the input variable of a new model

Weighted Average of $\{v_i\}$ – Type 1

- We search for the combination of weights that maximize the F Score. The optimal **simulated** weights are:

Random Forest	*LightGBM*	*XGBoost*	*CatBoost*	*GLM*
0.2	0.41	0.22	0.1	0.07

LightGBM

- We consider the $\{v_i\}$ and all the 26 original variables as input features for the Second Layer
- Then we use the Shap Value (see [7]) to identify the most important original features

XGBoost

- We do the same as of the previous model, except that we fit a new XGBoostModel

Type 2

Analysis of Results (1/4)

1. Starting from the most common **metrics** for classifier's evaluation

   - Accuracy = $(TP+TN)/(P+N)$
   - Error = $(FP+FN)/(P+N)$
   - Precision = $TP/(TP+FP)$
   - Recall/TP rate = TP/P
   - FP Rate = FP/N

Predicted class

Actual class	Pos	Neg	
Pos	*TP*	*FN*	*P*
Neg	*FP*	*TN*	*N*

2. And calculating a First Layer **Predictions Matrix Correlation**

	Random Forest	*LightGBM*	*CatBoost*	*XGBoost*	*GLM*
Random Forest	1	0.69	0.40	0.68	0.41
LightGBM		1	0.48	0.79	0.42
CatBoost			1	0.49	0.27
XGBoost				1	0.40
GLM					1

The **Tree Based Models**, such as Random Forest, LightGBM and XGBoost, tend to have **high correlations**

Analysis of Results (2/4)

We show in the following slides the **results** for each of the chosen model

Singular Models

Table 1: Train Out-Of-Fold Predictions of First Layer

ML Model	*Precision*	*Recall*	*F Score*
Random Forest	12.28%	32.02%	17.75%
LightGBM	13.08%	33.93%	18.89%
CatBoost	8.72%	31.54%	13.66%
XGBoost	12.90%	31.22%	18.26%
GLM	10.43%	21.74%	14.10%

Table 2: Test Predictions of First Layer

ML Model	*Precision*	*Recall*	*F Score*
Random Forest	12.37%	32.98%	17.99%
LightGBM	13.69%	34.03%	19.53%
CatBoost	10.05%	40.48%	16.09%
XGBoost	13.63%	30.88%	18.92%
GLM	11.03%	23.84%	15.08%

- The LightGBM and the XGBoost are the most performing singular ML models
- Observe how the out-of-fold statistics on the train set follow the same order of magnitude of the test set

Analysis of Results (3/4)

Two Layer Ensemble Model – Type 1

Best simulated results

Table 4: Weighted Average of First Layers ML Models

Random Forest	*LightGBM*	*XGBoost*	*CatBoost*	*GLM*	*Precision*	*Recall*	*F Score*
0.2	0.41	0.22	0.1	0.07	13.67%	31.47%	19.96%
0.28	0.42	0.19	0.07	0.04	13.70%	31.03%	19.02%
0.27	0.32	0.16	0.12	0.13	13.75%	30.77%	19.00%
0.28	0.52	0.03	0.13	0.04	13.10%	34.52%	19.00%
0.18	0.49	0.00	0.18	0.15	13.10%	34.52%	19.00%

- The results in Table 4 clearly show that the LighGBM is the most important model, followed by the Random Forest and the CatBoost
- Among the five best models there is not much difference in terms of performance
- While regarding the weights we learn that as the LightGBM and CatBoost increase in importance the XGBoost weights less → **Correlation**

Analysis of Results (4/4)

Two Layer Ensemble Model – Type 2

Table 5: Train Out-Of-Fold Predictions of Second Layer

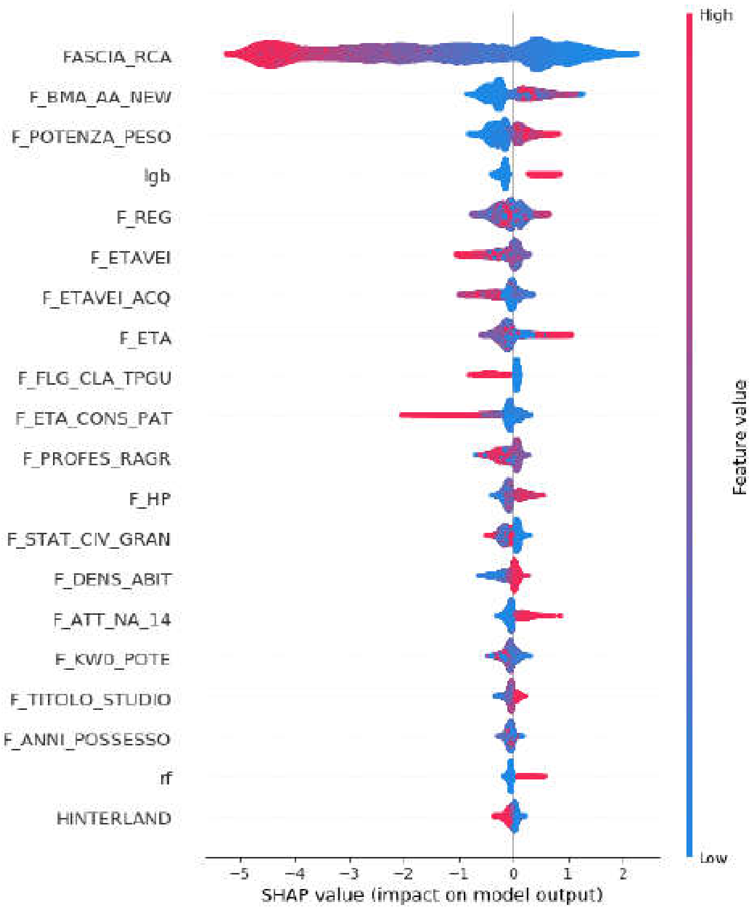
ML Model	*Precision*	*Recall*	*F Score*
XGBoost with all features	12.50%	28.76%	17.43%
XGBoost with best features	12.84%	30.70%	18.11%
LightGBM with all features	12.26%	33.30%	17.92%
LightGBM with best features	13.70%	32.68%	19.31%

Table 6: Test Predictions of Second Layer

ML Model	*Precision*	*Recall*	*F Score*
XGBoost with all features	13.52%	28.64%	18.36%
XGBoost with best features	14.40%	31.93%	19.87%
LightGBM with all features	13.65%	36.58%	19.88%
LightGBM with best features	14.01%	36.13%	20.19%

- In table 6 we present the final results of the Two Layer Ensemble Model on the test set
- The best performance is obtained by the LightGBM with **best feautures**, increasing our confidence in the model performance
- **Best features** are evaluated thanks to the **Shap Value** (see [7] for a deep discussion)

Focus on Shap Value



- It relies on solid Game Theory methodologies, i.e. the **Shapley Values**, where the n input variables are metaphorically equivalent to n players playing of a particular game
- Figure shows the most **important features**, order from the most significant to the least one and, analyzing the **colours**, it is possible to understand if a high level of the feature **impacts positive or negatively the probability of conversion**
- Chart is called "**violin plot**"
- Consider for example the lgb feature, that is the output of the first layer LighGBM, a high value of this feature, i.e. a probability of almost 1, produces a positive impact on the output of the model, as you may guess.
- The same for the opposite

References

[1]. Friedman, J.: Greedy Function Approximation: A gradient Boosting Machine. The Annals of Statistics 29(5), 1189–1232 (2001);

[2]. Friedman, J.: Stochastic Gradient Boosting. Journal Computational Statistics&Data Analysis - Nonlinear methods and data mining archive 38(4), 367–378 (2002);

[3]. Breiman, L.: Random Forests. Journal Machine Learning archive 45(1), 5–32 (2001);

[4]. McCullagh, P., Nelder, J.A.: Generalized Linear Models, Second edition. Chapman & Hall/CRC Monographs on Statistics & Applied Probability (1989);

[5]. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics, New York (2009).

[6]. Snoek J., Larochelle H., Adams R.: Practical Bayesian Optimization of Machine Learning Algorithms. arXiv:1206.2944v2 [stat.ML] (2012).

[7]. S. M. Lundberg, G. G. Erion, Su-In L.: Consistent Individualized Feature Attribution for Tree Ensembles. arXiv:1802.03888 [cs.LG] (2018)

[8]. T. Chen, C. Guestrin: XGBoost: A Scalable Tree Boosting System. arXiv:1603.02754 [cs.LG] (2016).

[9]. G. Ke, Q. Meng, T. Finley, T.Wang,W. Chen,W. Ma, Q. Ye, T. Liu: LightGBM: A Highly Efficient Gradient Boosting Decision Tree. 31st Conference on Neural Information Processing Systems (NIPS 2017).

[10]. L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, A. Gulin: CatBoost: unbiased boosting with categorical features. arXiv:1706.09516 [cs.LG] (2017);

[11]. J. Bergstra, D. Yamins, D. D. Cox: Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. Proceedings of the 30th International Conference on Machine Learning (Atlanta, 2013).

Thank you for listening

Lorenzo Invernizzi
Generali Italia
lorenzo.invernizzi@generiali.com

Vittorio Magatti (presenter)
Willis Towers Watson
vittorio.magatti@willistowerswatson.com

SECTION COLLOQUIUM 2019