InData Labs

# Explanation of machine learning models predictions

Anton Kulesh
Data Scientist at InData Labs

Datathon
Minsk, July 2019

# “Black Box”. The problem of interpretability

A **“black box”** is a model that accepts inputs and gives responses, but does not explain how they were received

# Interpretability. Basics

*Definition*: **Interpretability = Human understanding**

- Interpretable models
  - Linear Regression
  - Logistic Regression
  - Decision Tree
  - Generalized Linear Models
  - Generalized Additive Models
  - Rule-based approaches
  - etc.
- “Black-box” models
  - Gradient boosting
  - Random forest
  - SVMs
  - Deep neural networks
  - etc.

Hall, Phan, Ambati. Ideas on interpreting machine learning, 2017

# Interpretability. Basics

*Definition*: **Interpretability = Human understanding**

- Scope
  - Global interpretation
  - Local interpretation
- Application domain
  - Model-specific
  - Model-agnostic

Hall, Phan, Ambati. Ideas on interpreting machine learning, 2017

# Interpretability. Why is it important?

## 1. High-risk domains

- Healthcare
- Finance
- Judicial system
- Security

- It’s not enough just to give a prediction
- Decisions that the system makes should be **clear to the human**
- A System must answer not only the question “what?”, but also **“why?”**
- A System must be highly **credible**

# Example. Discriminative bias. COMPAS

Will the prisoner commit the crime?



Angwin, Larson, Mattu and Kirchner. Machine Bias: There's software used across the country to predict future criminals. And it's biased against blacks. ProPublica, 2016.

# Crimes in the U.S. 2016

## Table 21A

| | Total arrests | | | | | | Percent distribution[1] | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Race | | | | | | | | | | | |
| Offense charged | Total | White | Black or African American | American Indian or Alaska Native | Asian | Native Hawaiian or Other Pacific Islander | Total | White | Black or African American | American Indian or Alaska Native | Asian | Native Hawaiian or Other Pacific Islander |
| **TOTAL** | **8,421,481** | **5,858,330** | **2,263,112** | **171,185** | **103,244** | **25,610** | **100.0** | **69.6** | **26.9** | **2.0** | **1.2** | **0.3** |

https://ucr.fbi.gov/crime-in-the-u.s/2016/crime-in-the-u.s.-2016/topic-pages/tables/table-21

# Interpretability. Why is it important?

## 2. Regulatory compliance

- General Data Protection Regulation (GDPR[1])
- Fairness and transparency
- Non-discrimination

[..] processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision[2].

1. GDPR: https://gdpr-info.eu/
2. Art. 22: Automated individual decision-making, including profiling
3. Recital 71: Profiling
4. Goodman, Flaxman. European Union regulations on algorithmic decision-making and a “right to explanation”, 2016

# Interpretability. Why is it important?

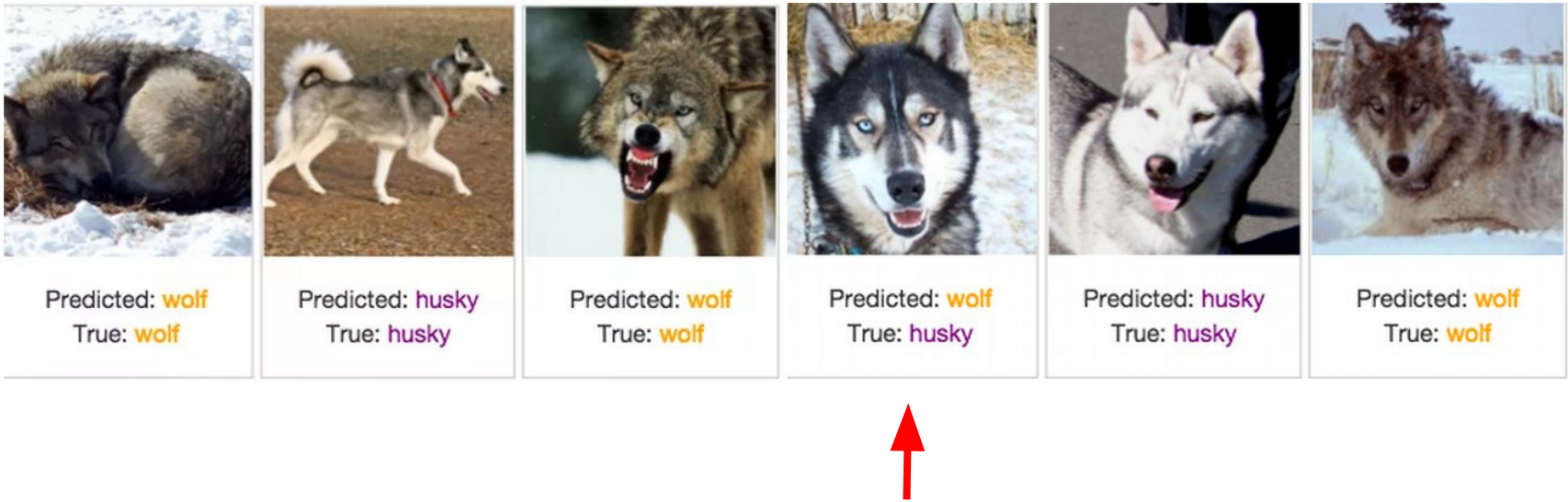
## 3. Model Improvement

- Debugging
- Feature re-engineering
- Model selection
- Identification of a **data leakage**
- **Dataset shift** detection

*[..] a **single metric**, such as classification accuracy, is an **incomplete description** of most real-world tasks*[1]

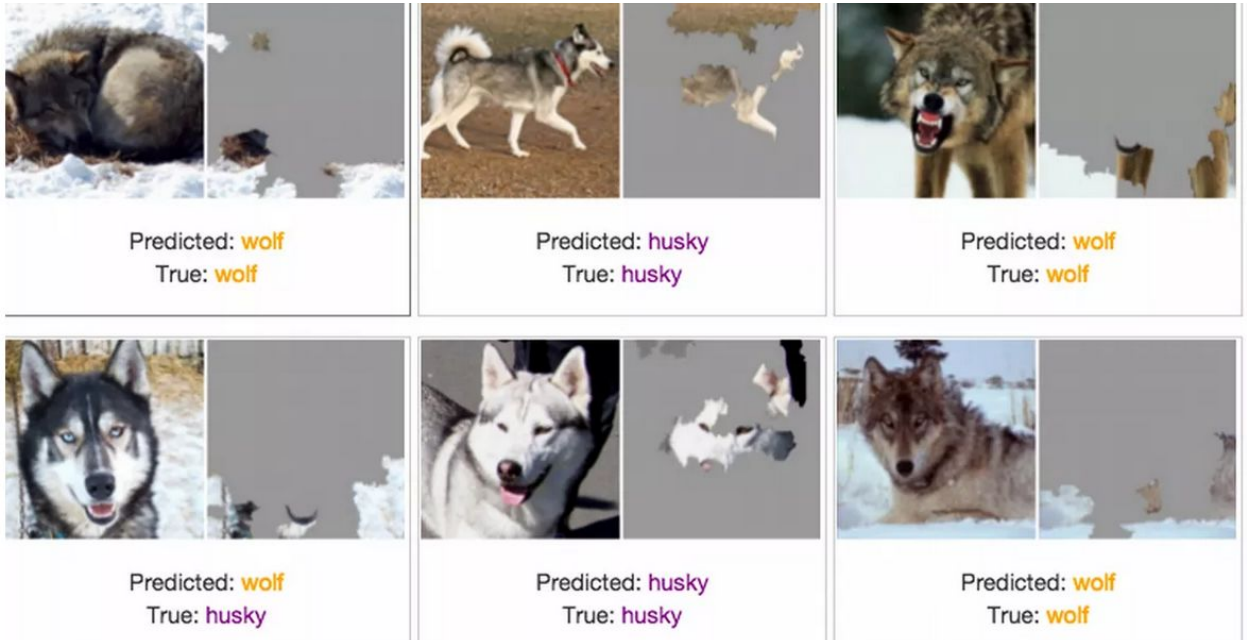
1. Doshi-Velez, Kim. Towards a rigorous science of interpretable machine learning, 2017

# Debugging. Poor generalization




1. Ribeiro, Singh, Guestrin. "Why should I trust you?: Explaining the predictions of any classifier", 2016.
2. Singh. Explaining Black-Box Machine Learning Predictions, 2017.

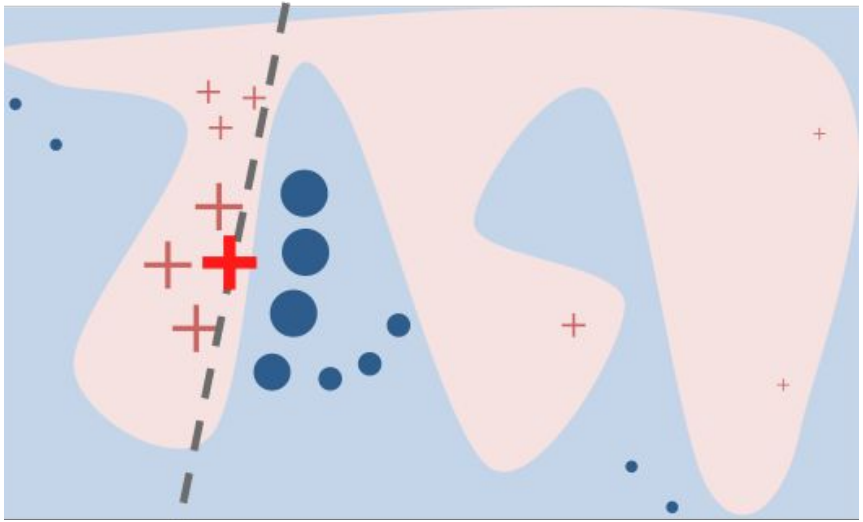
# Debugging. Poor generalization




1. Ribeiro, Singh, Guestrin. "Why should I trust you?: Explaining the predictions of any classifier", 2016.
2. Singh. Explaining Black-Box Machine Learning Predictions, 2017.

# LIME. Local Interpretable Model-agnostic Explanations

- LIME is an algorithm that can explain the predictions of **any** classifier or regressor in a faithful way, **by approximating it locally** with an interpretable model[1]
- Implementations
  - lime[2]
  - ELI5[3]



1. Ribeiro, Singh, Guestrin. "Why should I trust you?: Explaining the predictions of any classifier", 2016.
2. Implemented by the authors of the approach: https://github.com/marcotcr/lime
3. Alternative implementation: https://github.com/TeamHG-Memex/eli5

# LIME. Under the hood

To find the explanation function *g* we should minimize following objective function:

$$\xi = \operatorname*{arg\,min}_{g \in \mathcal{G}} \; L(f, g, \pi_{x'}) + \Omega(g).$$

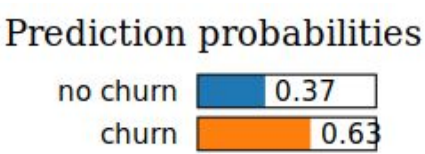
- *G* — class of interpretable models (e.g. linear functions)
- *L* — loss function (e.g. squared loss)
- *f* — black-box model (predictor)
- $\pi_{x'}$ — local kernel (e.g. exponential kernel)
- □(g) — regularization of model complexity (number of non-zero features)

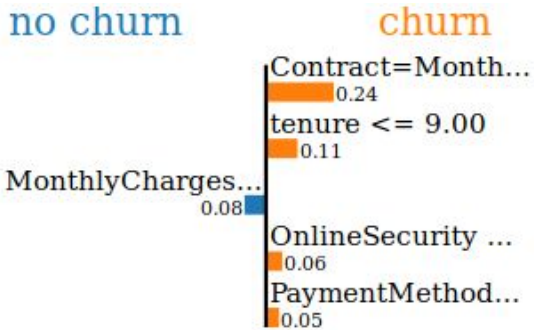
# LIME. Algorithm

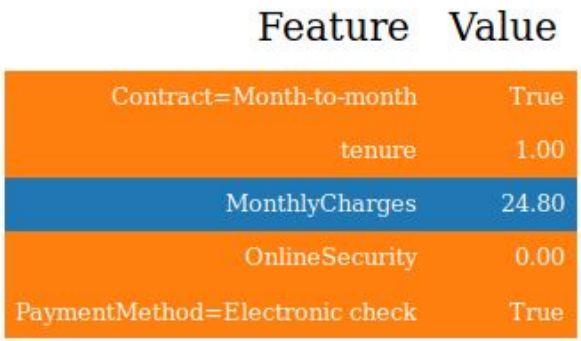
1. **Select** a sample (point) you want to explain
2. **Generate** new data samples based on selected sample and make predictions using original model
3. **Weight** new data points according to their proximity to the point of interest
4. **Train** a weighted, interpretable model on the obtained dataset
5. **Explain** the prediction by interpreting the local model

Nice short LIME explanation: Molnar. "Interpretable Machine Learning, 5.7. Local Surrogate (LIME)", 2019

# LIME. *Hands-on session




- **Problem**: binary classification (customer's churn prediction)
- **Model**: XGBoost
- **Library**: lime

*Link to the materials: https://github.com/akulesh/datathon-2019

# LIME. Pros and Cons

+ **Model-agnostic** (allows you to explain any model)
+ Working with **different data type** (tabular, text, images)
+ Submodular pick (method that selects a set of representative instances)
+ **Human-friendly** explanations

- Problems with **neighborhoods definition** (you have to try different kernel settings)
- **Robustness** (very close points can vary greatly in a simulated setting)
- Too time consuming for calculating global importances

1. Alvarez-Melis, Jaakkola. On the robustness of interpretability methods, 2018.
2. Nice short LIME explanation: Molnar. "Interpretable Machine Learning, 5.7. Local Surrogate (LIME)", 2019

# Additive Feature Attribution Methods. Definition

- The explanation function *g* is presented as linear combination by binary variables:

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

where $z' \in \{0, 1\}^M$, $M$ is the number of input features, and $\phi_i \in \mathbb{R}$.

1. Lundberg, Lee. A Unified Approach to Interpreting Model Predictions, 2017.

# Additive Feature Attribution Methods. Properties

- Local accuracy
  - The sum of the feature attributions is equal to the output of the function we are seeking to explain
- Missingness
  - Features that are already missing (such that $z_i=0$) are attributed no importance
- Consistency
  - Changing a model so a feature has a larger impact on the model will never decrease the attribution assigned to that feature

Only one possible explanation model *g* satisfies all properties...

1. Lundberg, Lee. A Unified Approach to Interpreting Model Predictions, 2017.
2. Lundberg, Erion, Lee. Consistent Individualized Feature Attribution for TreeEnsembles, 2019.

# SHAP. **SH**apley **A**dditive ex**P**lanation

SHAP[1] values combine conditional expectations of function given variables with the classic **Shapley values**[2] from game theory to attribute $\varphi_i$ values to each feature[3]:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} \left[ f_x(S \cup \{i\}) - f_x(S) \right],$$

- *f — original predictor,* $f_x(S) = E(f(x) \mid x_S)$ — expected value of the function conditioned on a subset S, *N — set of all input features S — subset of input features, M — number of input features*

1. Python implementation: https://github.com/slundberg/shap
2. Nice explanation of Shapley values: https://christophm.github.io/interpretable-ml-book/shapley.html
3. Lundberg, Erion, Lee. Consistent Individualized Feature Attribution for TreeEnsembles, 2019.

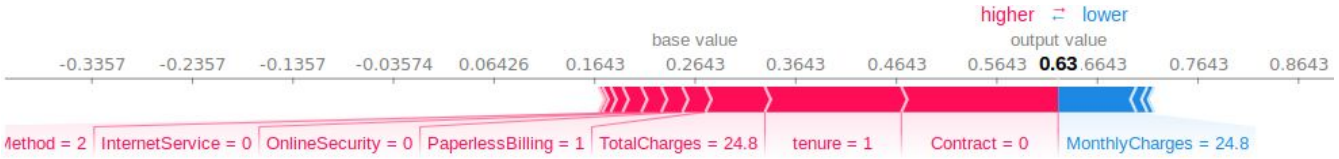
# SHAP. Modifications

- Model-Agnostic Approximations
  - Kernel SHAP (Linear LIME + Shapley values)
- Model-Specific Approximations
  - Linear SHAP (for linear models)
  - Low-Order SHAP
  - Max SHAP
  - Deep SHAP (DeepLIFT + Shapley values)
- **Tree SHAP**[1]
  - fast algorithm for tree models like XGBoost or Random Forest

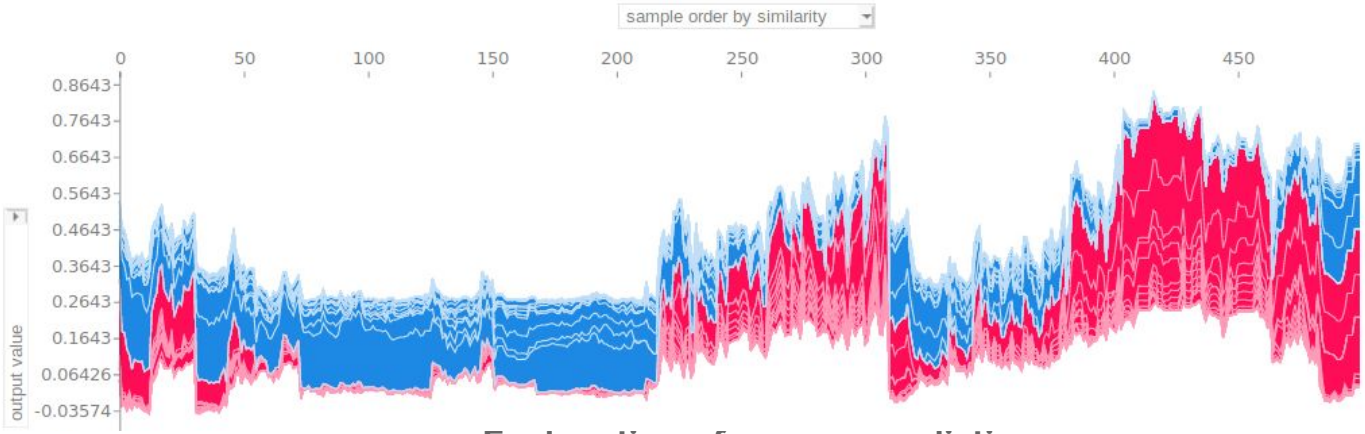
1. Lundberg, Erion, Lee. Consistent Individualized Feature Attribution for TreeEnsembles, 2019.
2. See many of examples of using SHAP here: https://github.com/slundberg/shap

InData Labs

# SHAP. *Hands-on session




**Explanation of a single prediction**

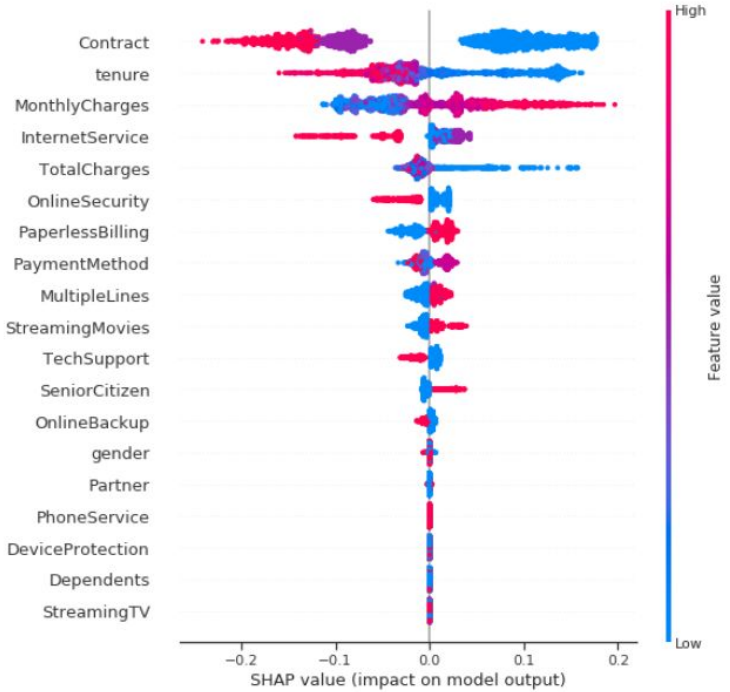


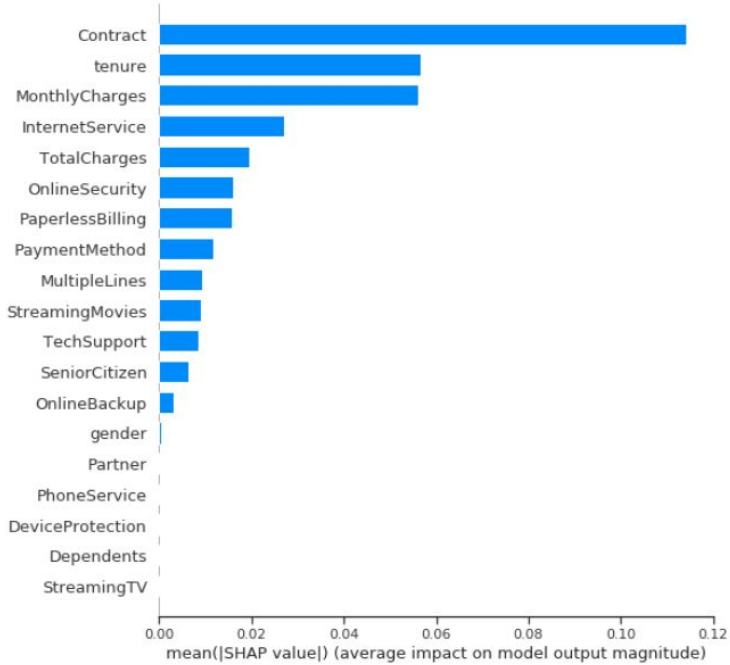

**Explanation of a many predictions**

- **Problem**: binary classification (customer's churn prediction)
- **Model**: XGBoost

*Link to the materials: https://github.com/akulesh/datathon-2019

# SHAP. Global explanations



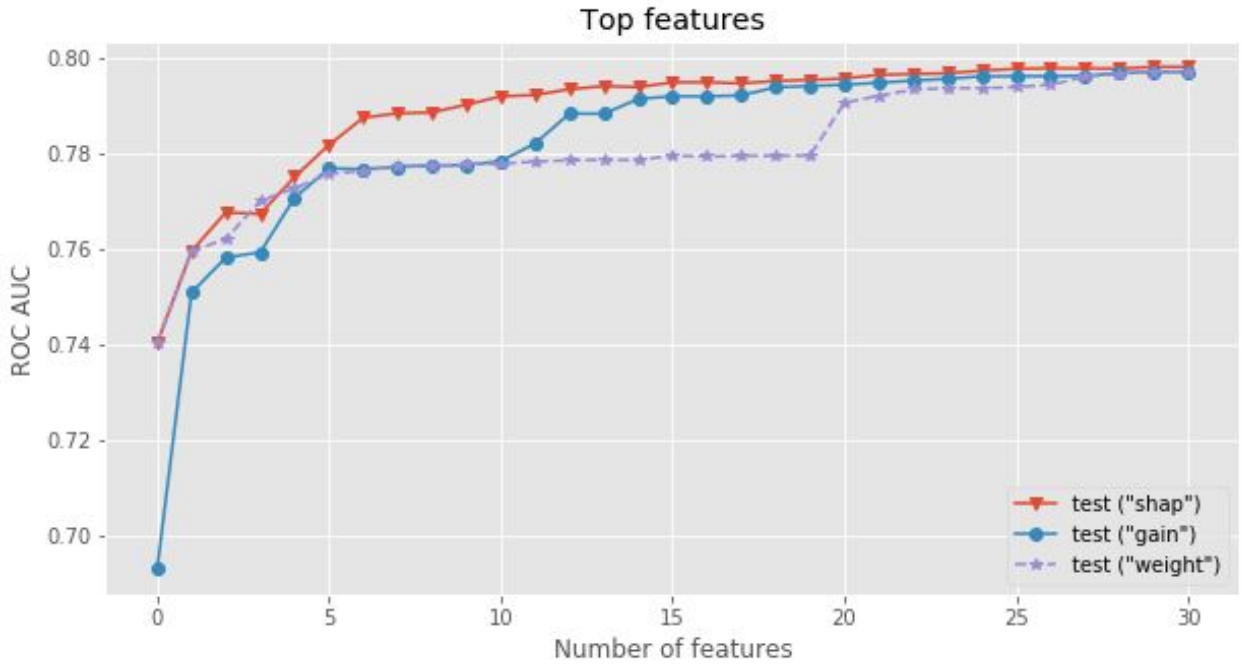



# Global feature importance

- Model-specific (tree ensembles)
  - **Gain**[1] — total reduction of loss or impurity contributed by all splits for a given feature
  - **Split count**[2] — how many times a feature is used to split
  - **Coverage** — he number of instances in tree node
- Model-agnostic
  - **Permutation** — randomly permute the values of a feature in the dataset and then observe how the model quality change
  - **SHAP** (mean(|shap_values|) — average magnitude of the individualized SHAP attributions

1. Breiman, Friedman and other. Classification and regression trees. CRC press, 1984.
2. Chen, Guestrin. XGBoost: A scalable tree boosting system. ACM, 2016.

# Feature importance. *Real case




* Here is a comparison of different feature importance techniques on real project data. We sorted features by importance (SHAP, “gain” and “weight”) and trained xgboost classifier on different feature subsets.

# SHAP. Pros and Cons

+ **Theoretical background**
  + based on Shapley values
  + all properties (local accuracy, missingness, consistency) are satisfied
+ **Tree SHAP**
+ Rich visualization
  + local explanation
  + global explanation
  + dependence of features
+ **Human-friendly** explanations

- Too **time consuming** (for model-agnostic case)
- Handling dependent features[1]
- Not really convenient to use for the analysis of texts (but the project is still in the active development phase)

1. Kjersti Aas, Martin Jullum, Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values, 2019.

# Out-of-scope

- Explanation of neural networks[4, 5]
- Attention-based RNNs[1, 2]
- Features/Model visualization[3, 6]
- Lots of tools...


1. Xu, Ba, Kiros and others. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, 2016.
2. Mullenbach, Wiegreffe, Duke and others. Explainable Prediction of Medical Codes from Clinical Text, 2018.
3. Cool resource with very nice material: https://distill.pub/
4. Lucid: A collection of infrastructure and tools for research in neural network interpretability.
5. Tao Lei, Regina Barzilay, Tommi Jaakkola. Rationalizing Neural Predictions, 2016.
6. Avanti Shrikumar, Peyton Greenside, Anshul Kundaje. Learning Important Features Through Propagating Activation Differences, 2017.
7. C. Molner. "Interpretable Machine Learning, 5.1. Partial Dependence Plot (PDP)", 2019

# Interpretability. Conclusion

With Interpretability:

- We can **understand decisions** of complex models more clearly
- We can **prevent overfitting** and access fairness of the model
- We can **improve** our model
- We can **choose** appropriate model for production
- We can **bring insights** for business
- ...

# Frameworks

| Tools | Notes |
|---|---|
| SHAP | Fair explanation of model predictions. Optimized for tree-based models (Tree SHAP). |
| lime | Works with tabular data, texts and images (original implementation). |
| ELI5 | Nice tool for inspecting “white boxes”. Also includes alternative implementation of LIME algorithm. |
| InterpretML | Open-source python package by Microsoft for training interpretable models and explaining black-box systems. Current version is quite raw but this tool looks very promising. |
| What-If Tool | New feature of the open-source TensorBoard web application, which let users analyze an ML model without writing code. See details in Google AI Blog |
| Lucid | A collection of infrastructure and tools for research in neural network interpretability. |
| Yellowbrick | Visual analysis and diagnostic tools to facilitate machine learning model selection. |

# Sources


- Christoph Molnar. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2019.
- Scott M. Lundberg, Su-In Lee. A Unified Approach to Interpreting Model Predictions, 2017.
- Scott M. Lundberg, Gabriel G. Erion, Su-In Lee. Consistent Individualized Feature Attribution for Tree Ensembles, 2018.
- Marco Tulio Ribeiro, Sameer Singh, Carlos Guestri. "Why Should I Trust You?" Explaining the Predictions of Any Classifier, 2016.
- Riccardo Guidotti, Anna Monreale, Franco Turini and others. A Survey Of Methods For Explaining Black Box Models, 2018.
- Finale Doshi-Velez, Been Kim. Towards A Rigorous Science of Interpretable Machine Learning, 2017.
- А. Дьяконов. Интерпретации чёрных ящиков, 2018.
- [video] K. Lemagnen. Open the Black Box: an Introduction to Model Interpretability with LIME and SHAP, 2018.
- [video] CVPR18: Tutorial: Part 1: Interpretable Machine Learning for Computer Vision, 2018
- Repos with relevant materials: awesome 1 and awesome 2

InData Labs

# Explanation of machine learning models predictions

Anton Kulesh, Data Scientist at InData Labs

**Github** repo with materials: https://github.com/akulesh/datathon-2019
**Email**: a_kulesh@indatalabs.com
**LinkedIn**: www.linkedin.com/in/anton-kulesh

Datathon
Minsk, July 2019