Explaining Complex Machine Learning Models with LIME

Dr. Shirin Glander

shirin.glander@codecentric.de

2017-12-11

twimlai.com

TWiML & AI

Talk #7 Carlos Guestrin – Explaining the Predictions of Machine Learning models

Marco Ribeiro,
Sameer Singh &
Carlos Guestrin:
"Why should I trust you? Explaining the predictions of any classifier."
CoRR 2016

LIME

Trust

Transparency

Public perception

Improvement through feedback*

* e.g. to check feature quality

making informed decisions

Trade-off

Simple models won't be able to solve complex problems!

e.g. fit a linear model

"why does a model say that the patient has cancer?"

LIME: fit simple explanations
↳ coarse approximations to the underlying decision

⇨ explanations are found for individual cases by comparing them with similar cases

▫ find most important variables for specific predictions

human interpretable explanations

text data, image data, etc.

○ for neural networks, random forests, boosted decision trees, etc.

github.com/marcotcr/lime Python

recreated for R: github.com/thomasp85/lime

debug ML

Local

Interpretable

Model-Agnostic

Explanations

ML = Black Box

make use of human intuition

datascepetic.com

Trusting Machine Learning Models with LIME

DATA SKEPTIC Podcast Aug. 2016

"We typically over-estimate the accuracy of our ML models"

Explanations are done on original (input) features

Learn another model on top of the ML model

⇩

generate "new data" by permutation of the input data (test set)

⇩

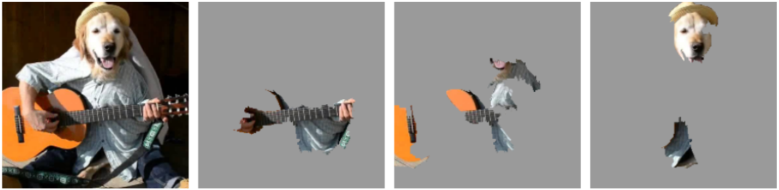
predictions on this new data with simple model → Local fit (approximation)

more weight is being put on data that is similar to the original data

Trust that the model bases decisions on features that make intuitive sense.

LIME can explain any classifier

- image recognition



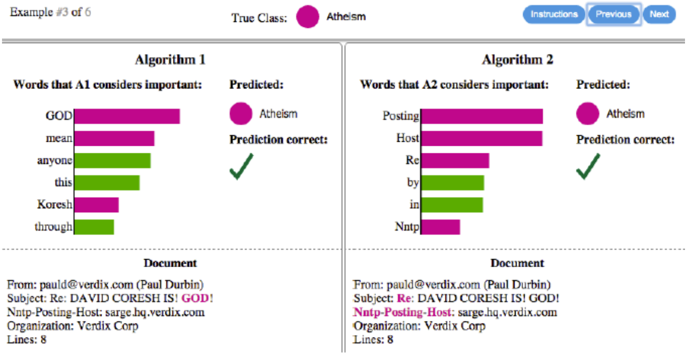
(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

Ribeiro, Singh, and Guestrin (2016)

LIME can explain any classifier

- text classification



Ribeiro, Singh, and Guestrin (2016)

How LIME works

1. Permutation of each test case to explain
2. Complex model predicts all permuted test cases
3. Distance between permutations and original text case is calculated and converted to similarity scores
4. Subsetting features with highest importance in complex model for each permuted test case
5. Fitting a linear model with the subsetted features to the permuted data (weights represent similarity score)
6. Using simple model to explain test case prediction

An example in R

- Data: Chronic Kidney Disease (http://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease)
- Nonparametric Missing Value Imputation using Random Forest (library(missForest))
- Categorical features converted to dummy variables (library(dummies))
- Scaled and centered

Predictor: ckd or notckd (class)

- Random Forest model with library(caret) (5x10 repeated CV)

The model

```
## Random Forest
##
## 360 samples
##  48 predictor
##   2 classes: 'ckd', 'notckd'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 324, 324, 324, 324, 325, 324, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9922647  0.9838466
##   25    0.9917392  0.9826070
##   48    0.9872930  0.9729881
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```

Predictions

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ckd notckd
##     ckd     23      2
##     notckd   0     15
##
##                Accuracy : 0.95
##                  95% CI : (0.8308, 0.9939)
##     No Information Rate : 0.575
##     P-Value [Acc > NIR] : 1.113e-07
##
##                   Kappa : 0.8961
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 1.0000
##             Specificity : 0.8824
##          Pos Pred Value : 0.9200
##          Neg Pred Value : 1.0000
```

Explaining the predictions

Explanation function:

- train_x is the training data
- model_rf is the complex model
- n_bins = 10 groups continuous variables into 10 bins
- quantile_bins = TRUE bases bins on quantiles (bins are not evenly spread across data range)
- dist_fun = “euclidean” sets distance function to calculate weights

```
library(lime)
explainer <- lime(train_x,
            model_rf,
            n_bins = 10,
            quantile_bins = TRUE,
            dist_fun = "euclidean")
```

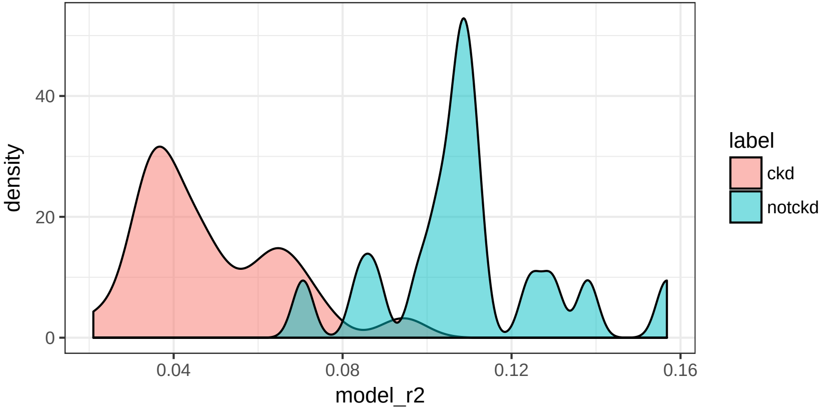
Explaining the predictions

- n_labels = 1 because we want to explain the most likely class predicted
- n_features = 8 returns top 8 most important features for each test case
- n_permutations = 1000 permutes test case 1000x
- feature_select = “highest_weights”:fits a ridge regression and selects the top features with highest absolute weight

```r
explanation_df <- lime::explain(test_x,
                    explainer,
                    n_labels = 1,
                    n_features = 8,
                    n_permutations = 1000,
                    feature_select = "highest_weights")
```

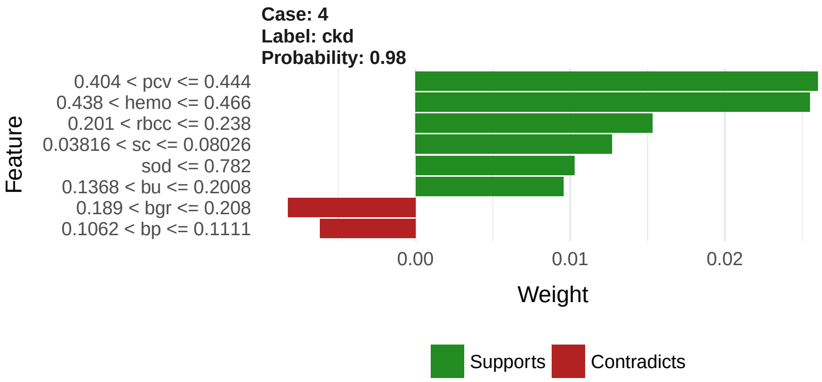
Explanation quality

- model r^2



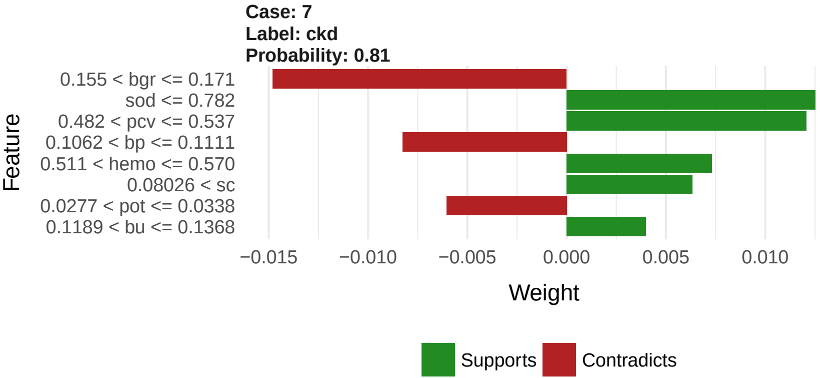
Plotting the explanations

```
plot_features(explanation_df[1:8, ])
```



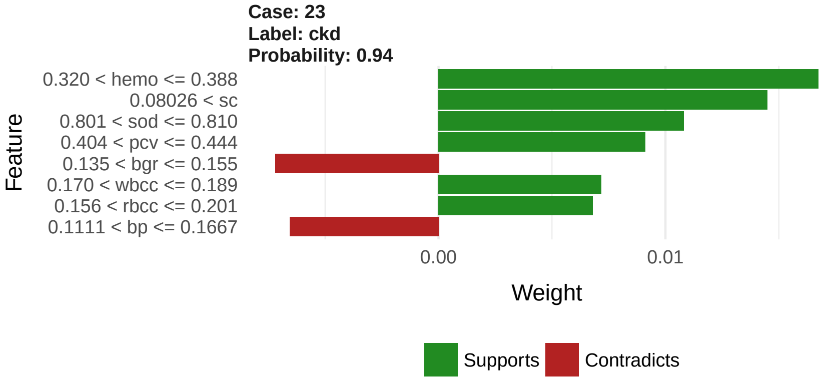
Plotting the predictions

```
plot_features(explanation_df[9:16, ])
```



Plotting the predictions

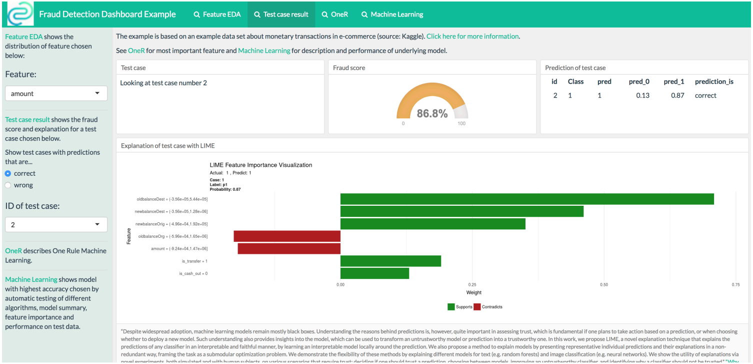
```r
plot_features(explanation_df[17:24, ])
```



LIME in action

- Explaining fraud predictions:

https://shiring.shinyapps.io/fraud_example_dashboard/



More about LIME

Publication

- Ribeiro, Singh, and Guestrin (2016)

Contribute

- https://github.com/marcotcr/lime
- https://github.com/thomasp85/lime

Thank you!

...and stay connected...

You can find me on

- my blog: www.shirin-glander.de
- Twitter: https://twitter.com/ShirinGlander
- Github: https://github.com/ShirinG

Code and slides will go up on my blog!

MünsteR User group

- https://www.meetup.com/Munster-R-Users-Group

Ribeiro, Marco Túlio, Sameer Singh, and Carlos Guestrin. 2016. "'Why Should I Trust You?': Explaining the Predictions of Any Classifier." CoRR abs/1602.04938. http://arxiv.org/abs/1602.04938.