Interpretable Machine Learning

JARON ARBET

Predictive Modeling

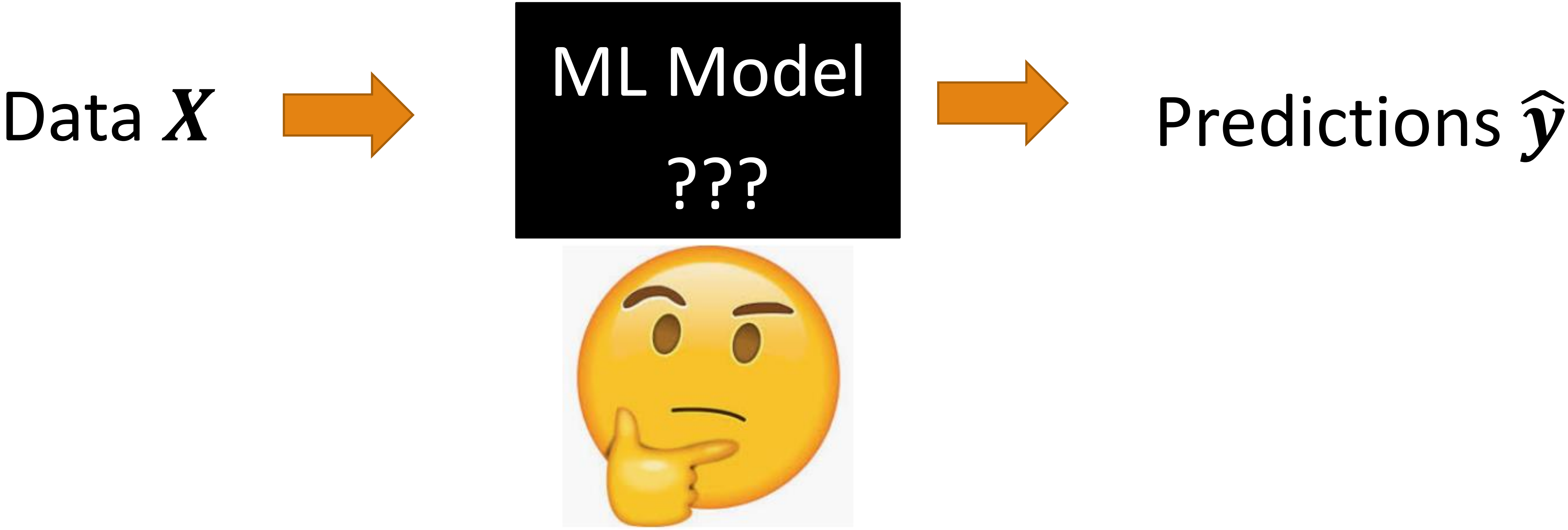
$$\boldsymbol{Y = f(X) + e}$$

- Assume outcome "$\boldsymbol{Y}$", can be predicted as a function "$\boldsymbol{f}$" of measured features "$\boldsymbol{X}$" + error

- **Classical models** (e.g. GLM) assume each feature has a *linear* and *additive* relationship with $\boldsymbol{Y}$ (i.e. no interactions), and N > P.
  - Easy to interpret, but probably unrealistic in many applications

- **Machine Learning** allows for more complex/flexible relationships between $\boldsymbol{X}$ and $\boldsymbol{Y}$. Random forests, SVM, MARS, neural nets, can automatically allow for complex non-linear and interaction effects for any predictor, allow P > N.

Although machine learning can often produce more accurate predictions, the price is that they are usually much harder to interpret



Data $\boldsymbol{X}$

ML Model

???

Predictions $\hat{\boldsymbol{y}}$

iml R package: “interpretable machine learning”

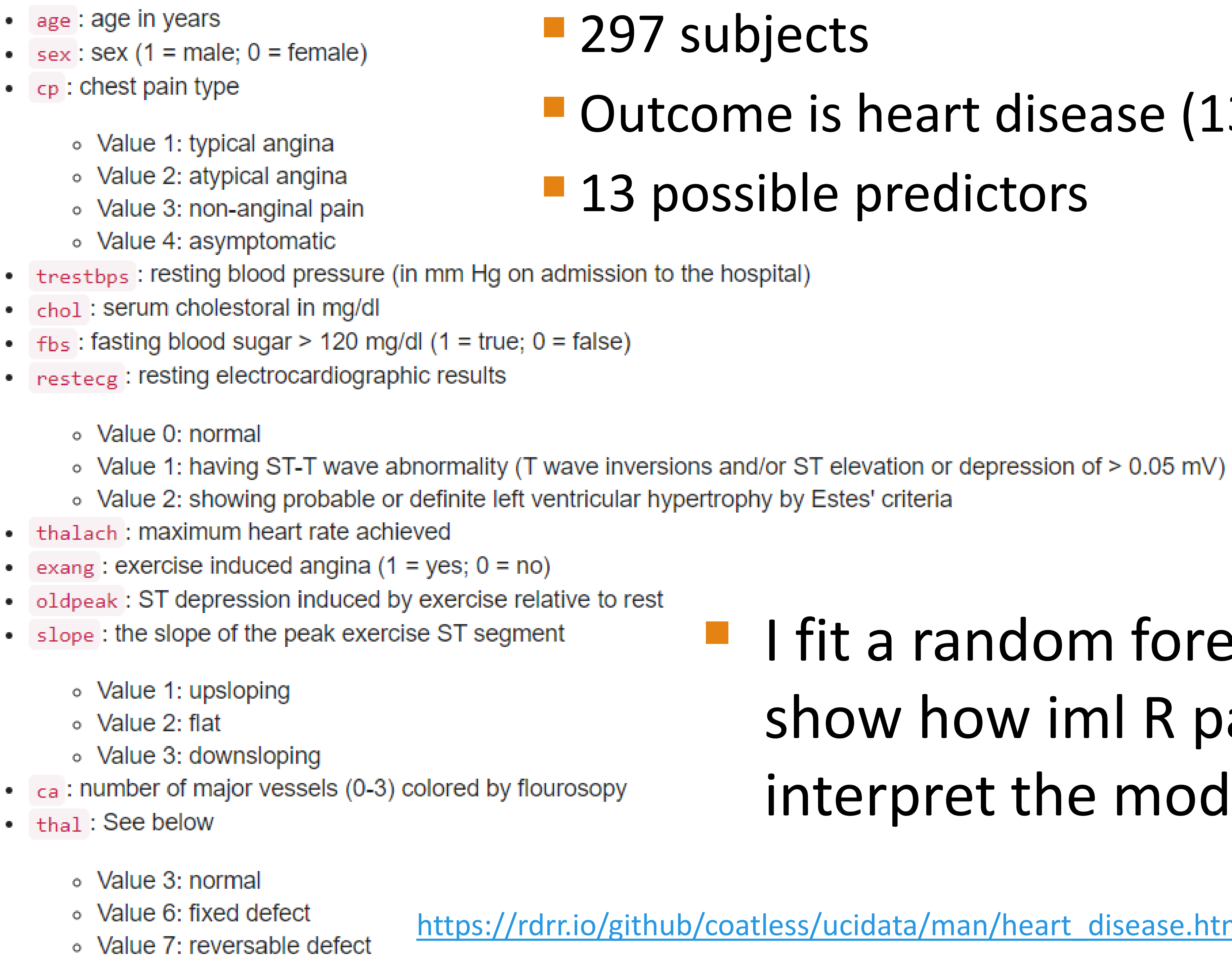
- https://cran.r-project.org/web/packages/iml/index.html
- Tutorial: https://cran.r-project.org/web/packages/iml/vignettes/intro.html
- Free book: https://christophm.github.io/interpretable-ml-book/
- Supports any ML model from the caret R package (>200 models)

Implements many state of the art methods for interpreting ML models:

- **Visualize relationships btwn X and Y** (partial dependence plots, ICE plots)
- **Variable Importance scores**
- **Interaction scores**: identify predictors that interact
- **LIME:** explain how a ML model makes a prediction for a given subject
- **Shapley Values:** uses game theory to explain how a prediction is made

Example: Heart Disease study

- 297 subjects
- Outcome is heart disease (137 have, 160 do not)
- 13 possible predictors



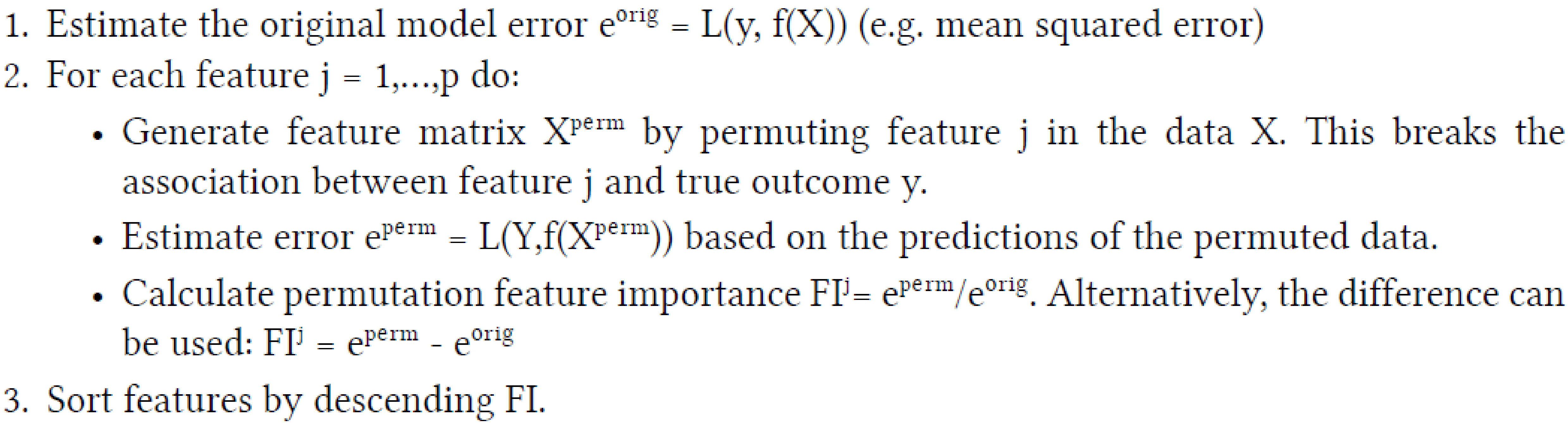
- `age` : age in years
- `sex` : sex (1 = male; 0 = female)
- `cp` : chest pain type
  - Value 1: typical angina
  - Value 2: atypical angina
  - Value 3: non-anginal pain
  - Value 4: asymptomatic
- `trestbps` : resting blood pressure (in mm Hg on admission to the hospital)
- `chol` : serum cholestoral in mg/dl
- `fbs` : fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
- `restecg` : resting electrocardiographic results
  - Value 0: normal
  - Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
  - Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
- `thalach` : maximum heart rate achieved
- `exang` : exercise induced angina (1 = yes; 0 = no)
- `oldpeak` : ST depression induced by exercise relative to rest
- `slope` : the slope of the peak exercise ST segment
  - Value 1: upsloping
  - Value 2: flat
  - Value 3: downsloping
- `ca` : number of major vessels (0-3) colored by flourosopy
- `thal` : See below
  - Value 3: normal
  - Value 6: fixed defect
  - Value 7: reversable defect

- I fit a random forest model and will show how iml R package can help interpret the model

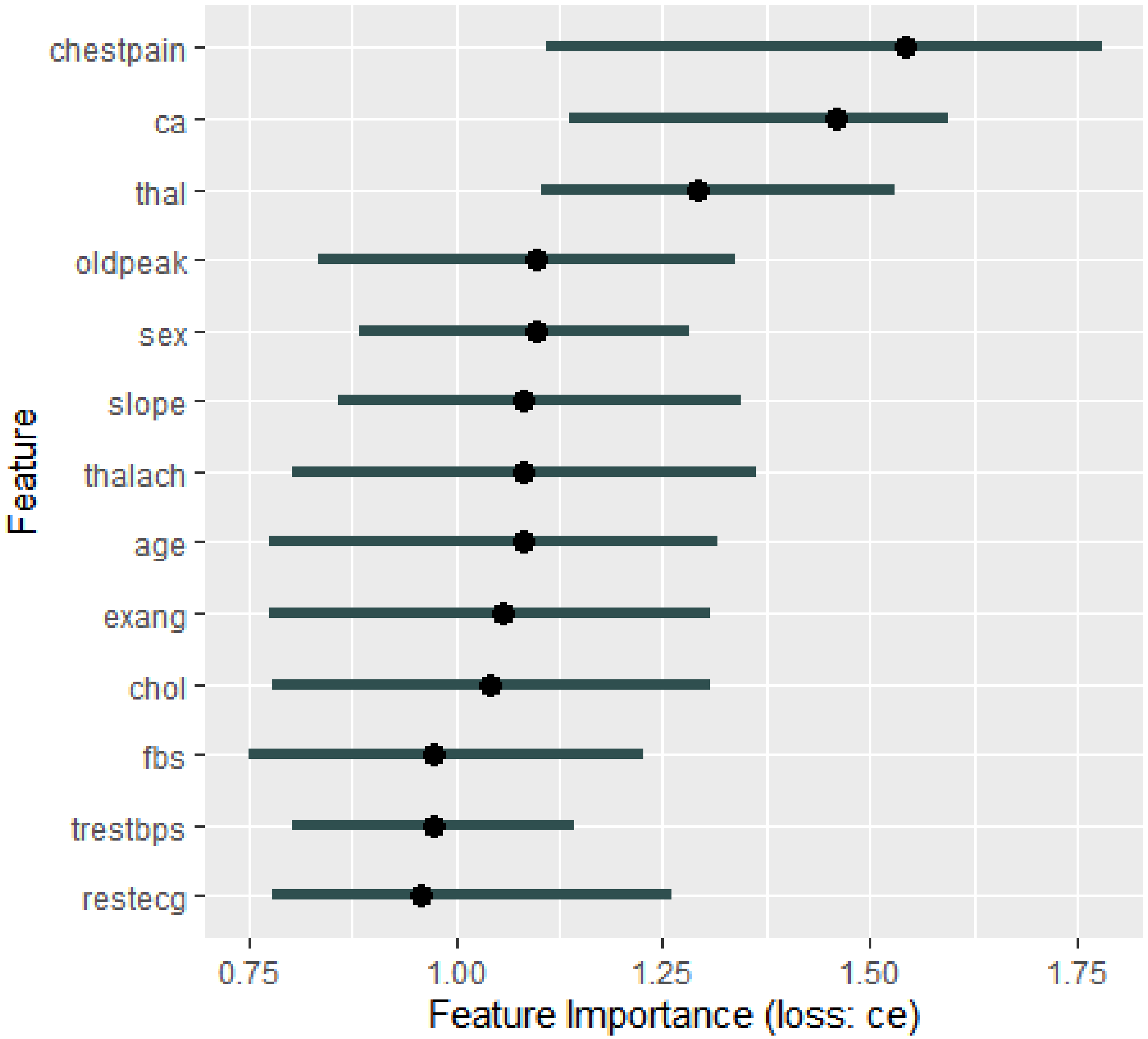
https://rdrr.io/github/coatless/ucidata/man/heart_disease.html

Variable Importance

- How important is each variable in predicting heart disease status?
- Permutation-based method



1. Estimate the original model error $e^{orig} = L(y, f(X))$ (e.g. mean squared error)
2. For each feature $j = 1,\ldots,p$ do:
   - Generate feature matrix X^{perm} by permuting feature j in the data X. This breaks the association between feature j and true outcome y.
   - Estimate error $e^{perm} = L(Y,f(X^{perm}))$ based on the predictions of the permuted data.
   - Calculate permutation feature importance $FI^j= e^{perm}/e^{orig}$. Alternatively, the difference can be used: $FI^j = e^{perm} - e^{orig}$
3. Sort features by descending FI.

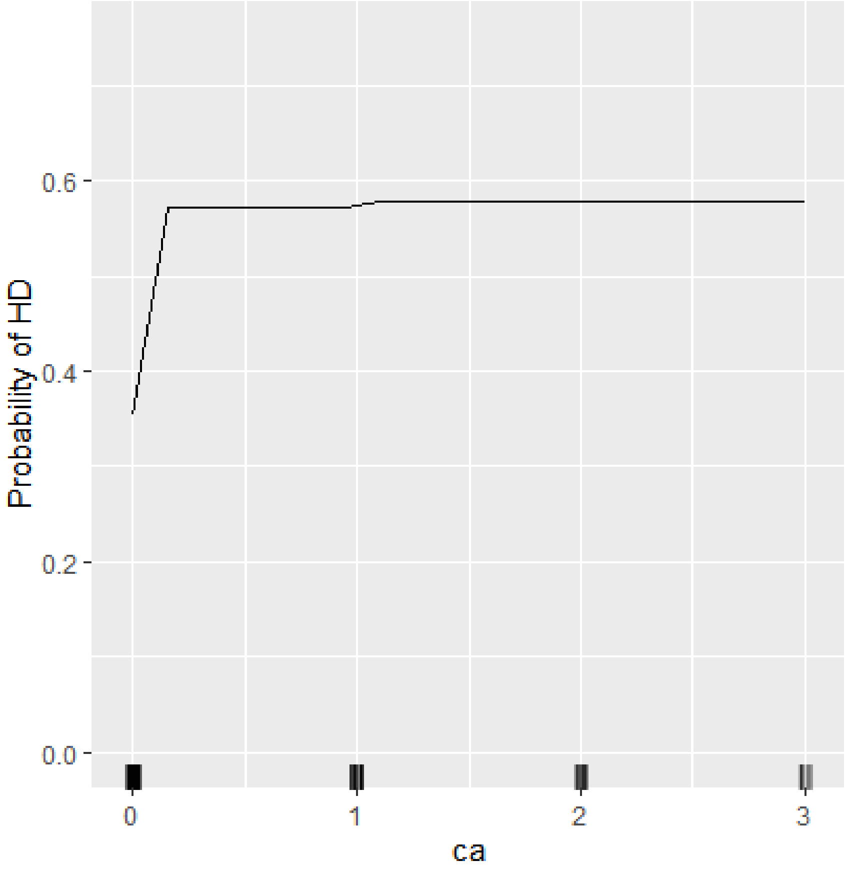


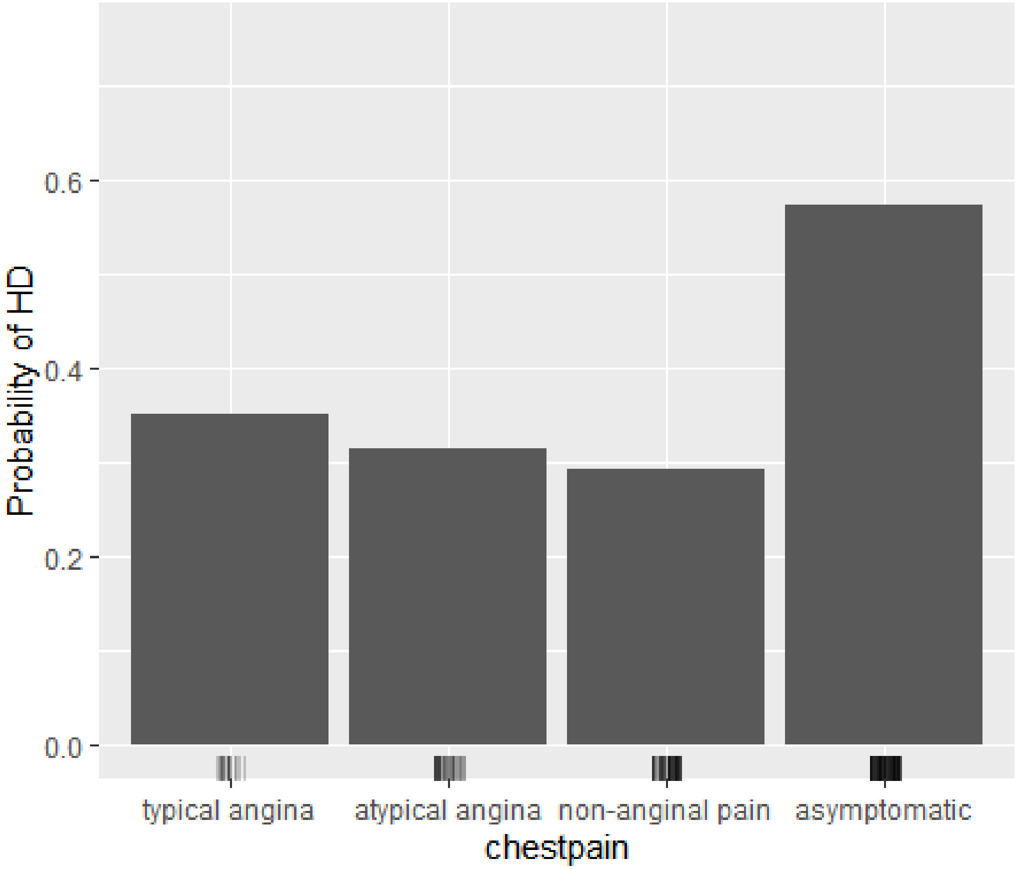
$$FI^j = e^{perm}/e^{orig}.$$

- FI near 1 means predictor is not important
- FI for chestpain=1.54, the prediction error increased 54% after permuting chestpain.

Visualize Effects

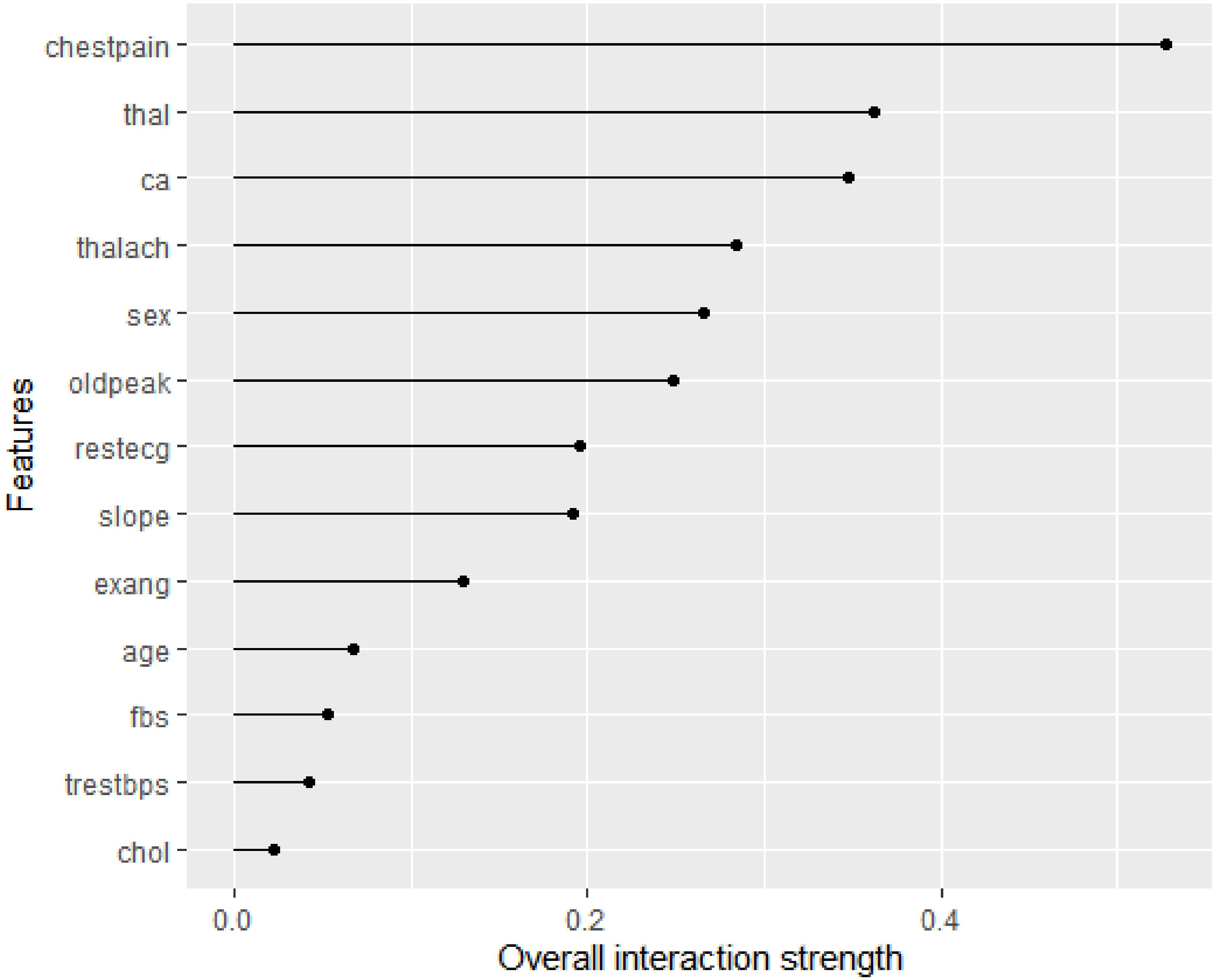
- “partial dependence plots” (**Friedman 2001**): can be used to visualize the relationship between $\boldsymbol{Y}$ and a predictor $\boldsymbol{X_j}$
- Similar to “marginal effect plots” (calculate $\hat{Y}$ for all values of $\boldsymbol{X_j}$ while holding all other predictors at their average value)





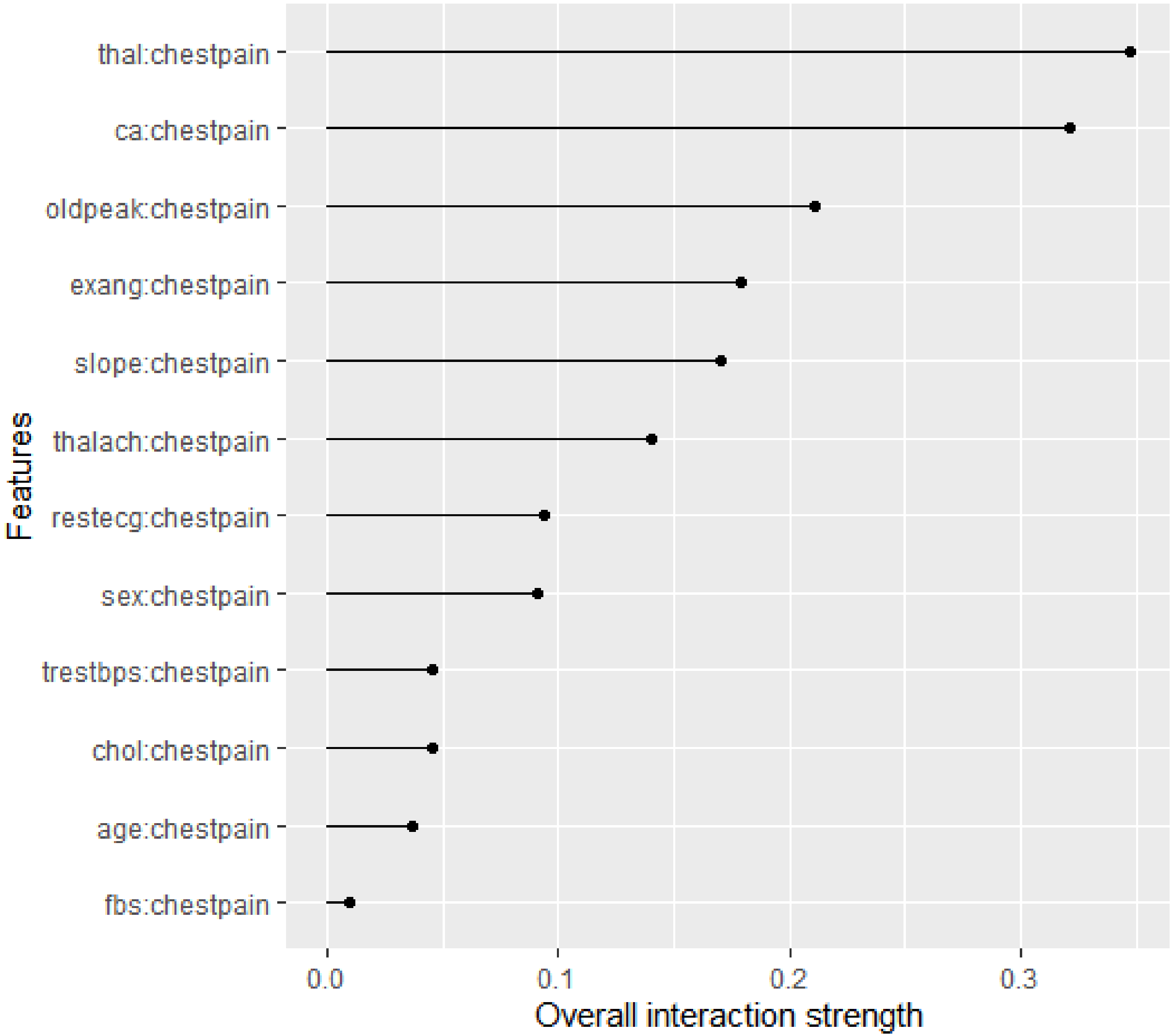
Interactions

- Friedman's "**H-statistic**" (**Friedman 2008**), 2 commonly used versions:
    1. Measure the interaction strength between 2 variables X_j and X_k (% of variance in the 2-dim partial dependence function of X_j, X_k with Y that is due to the interaction of X_j and X_k)
    2. Overall measure of interaction strength for a single variable X_j (% of variance in prediction function $\hat{f}$ that is due to ANY interaction effects involving X_j)
- H ranges from 0 to 1, with 0 meaning no interaction and larger values indicate stronger interaction effects

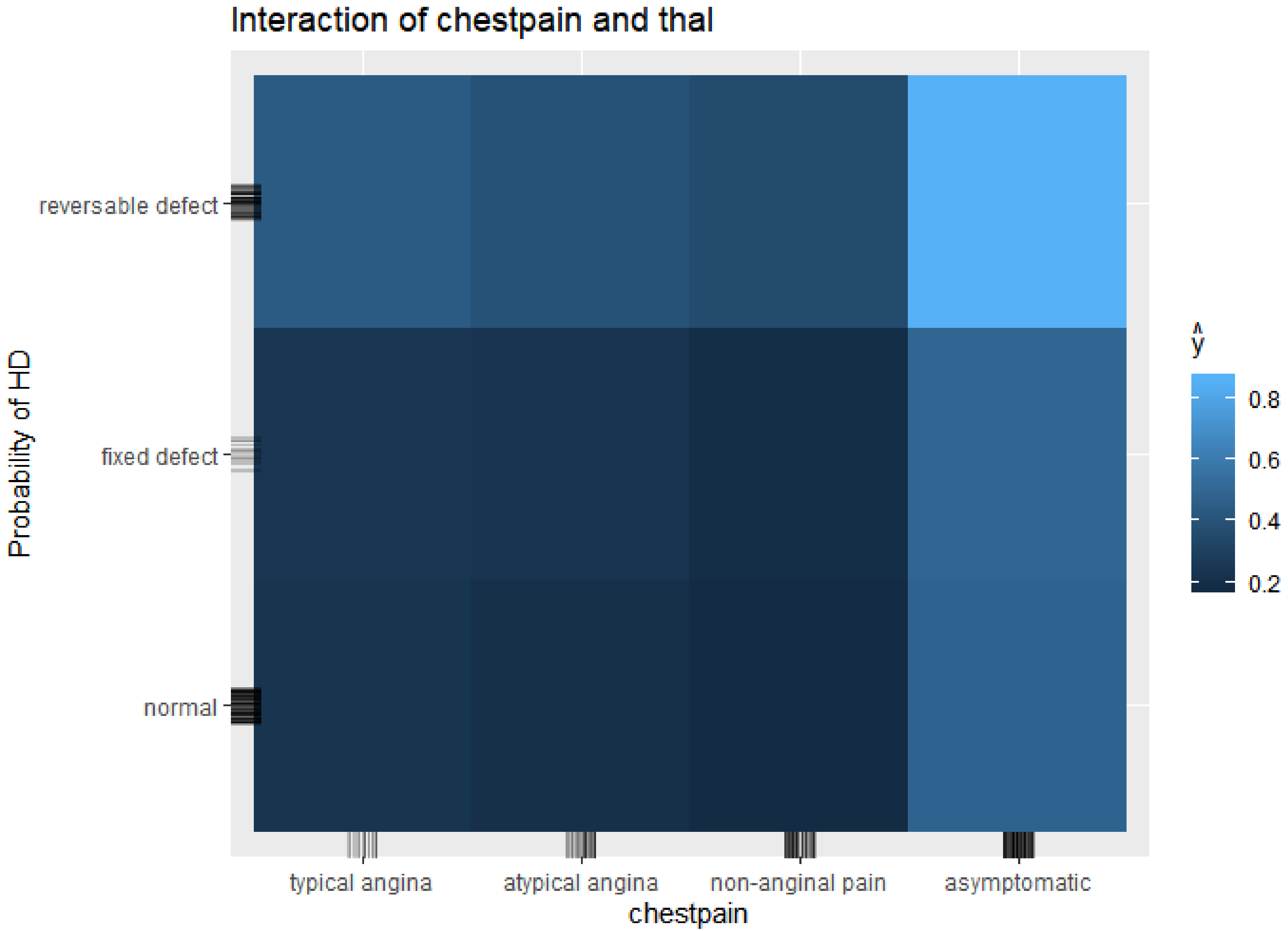


- 53% of the variance in the predictive function $\hat{f}$ is due to interaction effects involving chestpain
- Thal and ca also have fairly large interaction effects

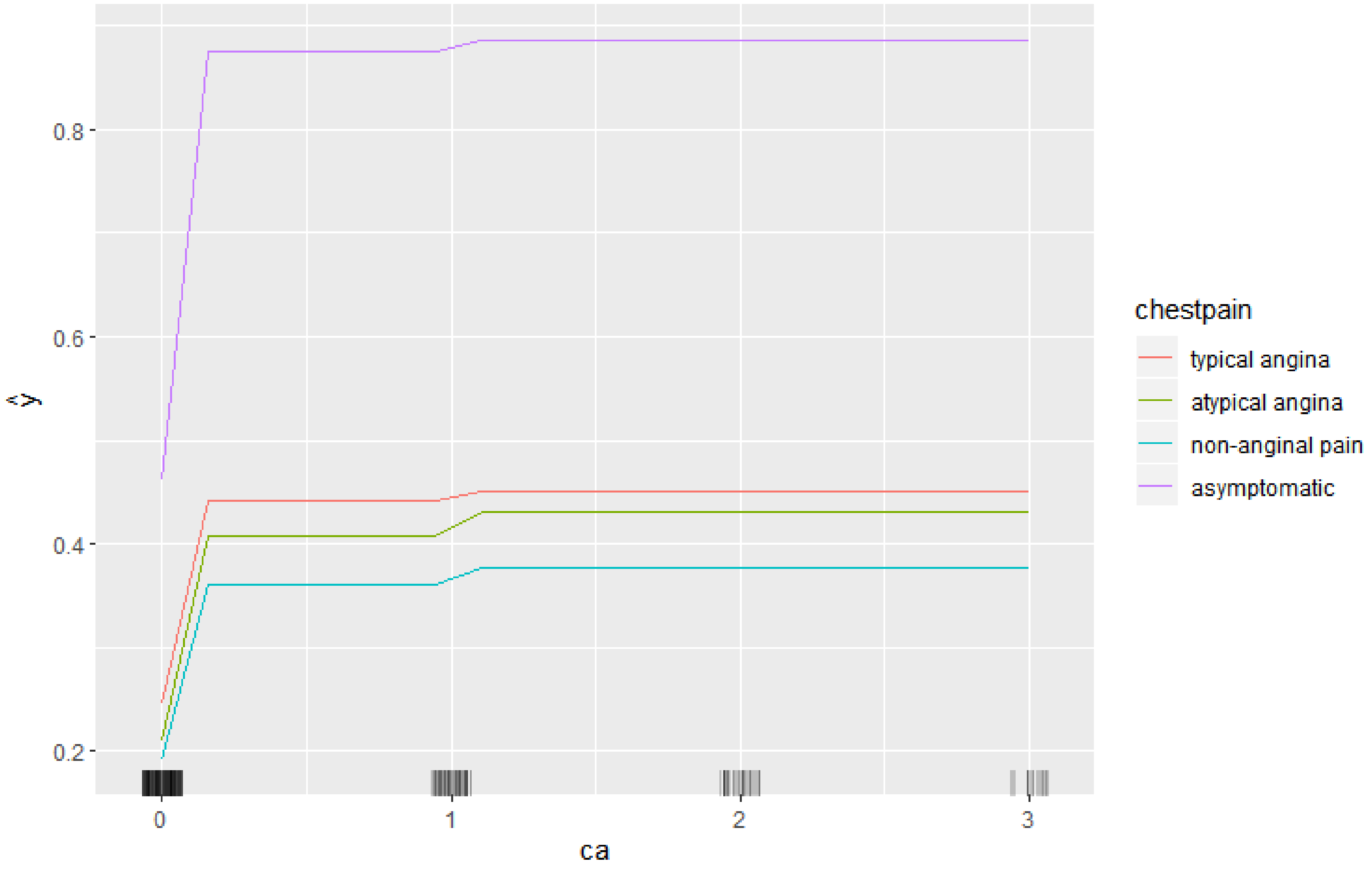
All 2-way interaction effects with chestpain



2-Dim partial dependence plots can then be used to visualize interaction effects



Interaction of chestpain and ca



LIME: “Local Interpretable model explanations”

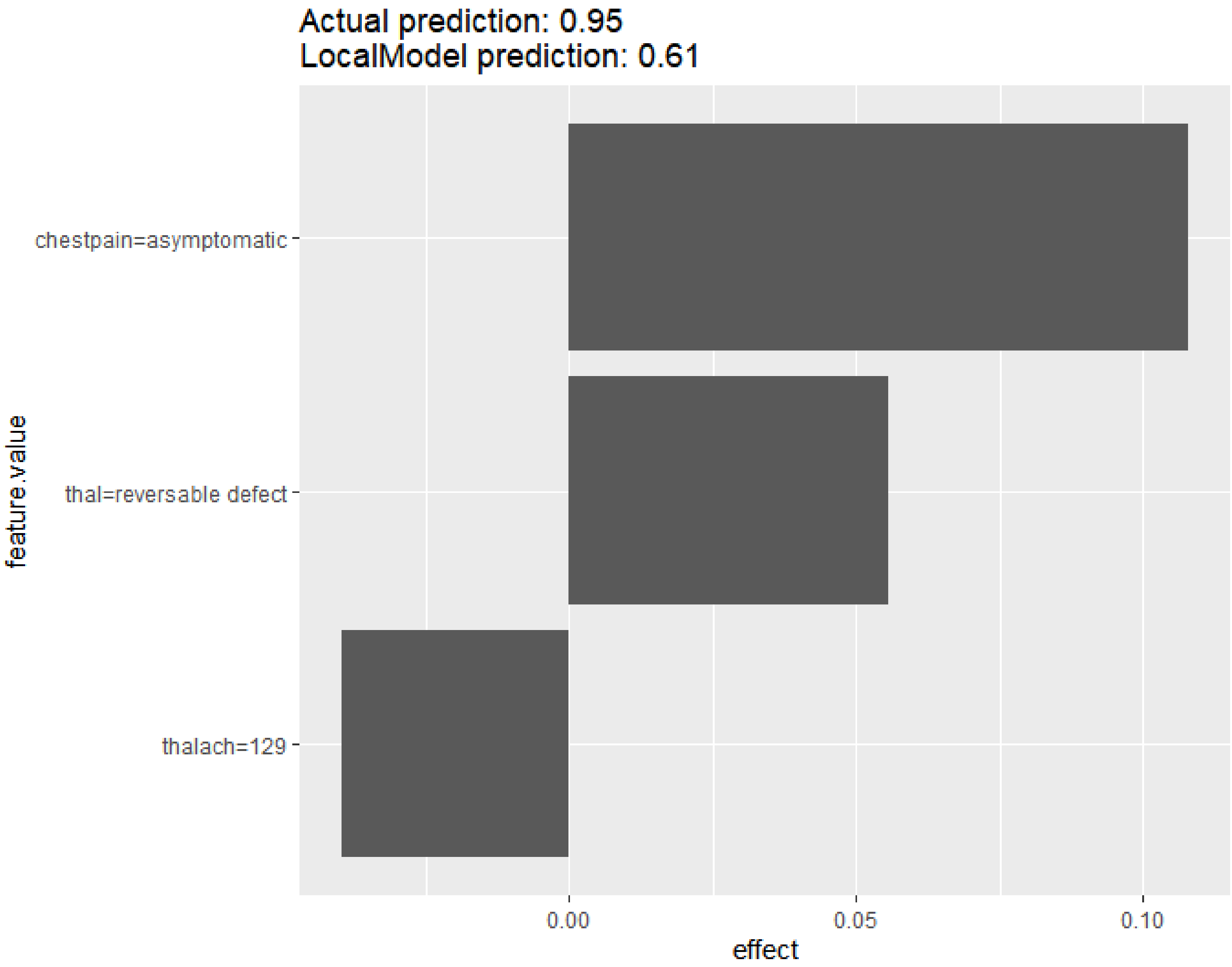
- Tulio Ribeiro 2016: “‘Why Should I Trust You?’ Explaining the Predictions of Any Classifier”

- **Goal:** explain why a black box ML model made the prediction it did for a particular subject

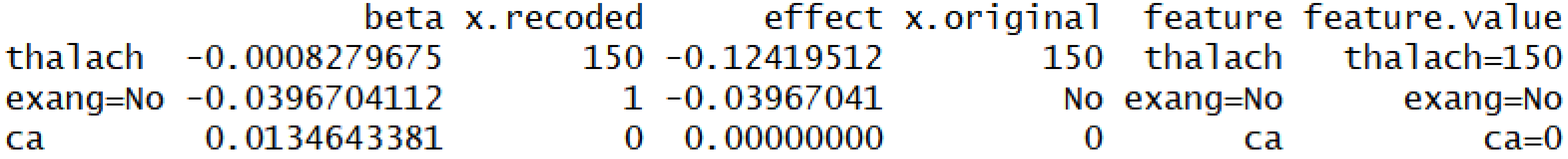
- Use simpler more interpretable models (e.g. linear regression, logistic regression) *locally* to explain how the subject’s feature values affected their prediction

- **Local?** Use a distance/similarity function to *weigh* all subjects in your dataset by how close they are to the subject of interest. Then fit a weighted linear/logistic regression model.

Here logistic regression is used with the top 3 predictors (chosen by Lasso)

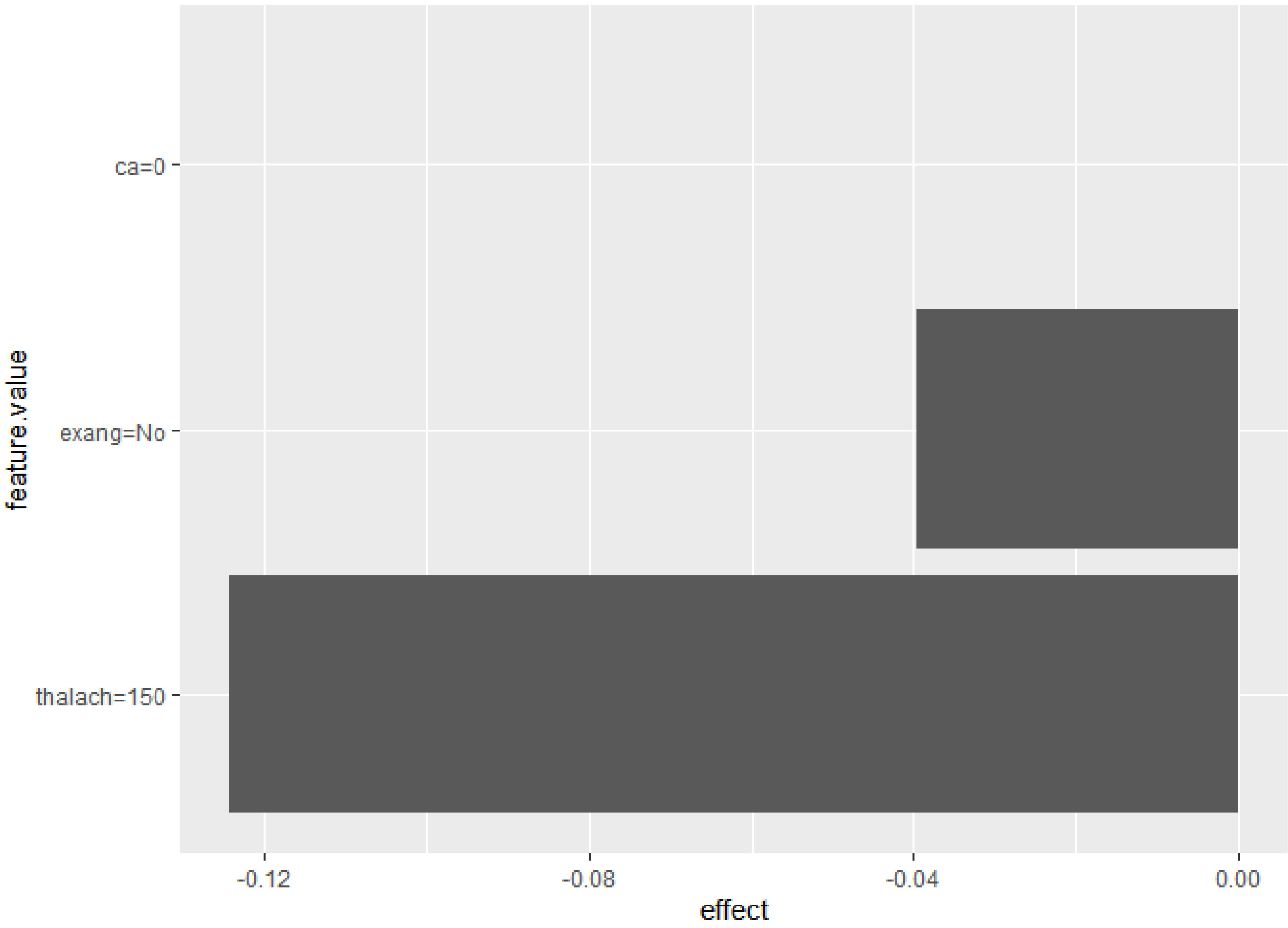


- Y-axis shows the feature values for this subject
- X-axis shows how the subject’s feature values affected their log-odds of having HD



```
                    beta x.recoded      effect x.original  feature feature.value
thalach  -0.0008279675       150 -0.12419512        150  thalach   thalach=150
exang=No -0.0396704112         1 -0.03967041         No exang=No      exang=No
ca        0.0134643381         0  0.00000000          0       ca          ca=0
```

Actual prediction: 0.34
LocalModel prediction: 0.44



References

- Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232
- Friedman, Jerome H., and Bogdan E. Popescu. "Predictive learning via rule ensembles." The Annals of Applied Statistics 2.3 (2008): 916-954.
- Molnar, Christoph. Interpretable machine learning. Lulu. com, 2019.
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should I trust you?" Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.