CREDIT SCORING
project summary

Nguyễn Ngọc Biển- Data Scientist, P. PTDL, TTCN
August, 2020

Nội dung

01 **Chọn dữ liệu**

Giới thiệu tổng quan về dự án và vấn đề cần giải quyết

02 **Lựa chọn metrics**

Lựa chọn metric. Một số đặc tính của metric AUC và GINI

03 **Features extraction**

Tổng quan về các kỹ thuật.

04 **Modelling**

Lựa chọn và tinh chỉnh mô hình

05 **Model Visualization**

Phương pháp visualize model và output

Chọn dữ liệu

1

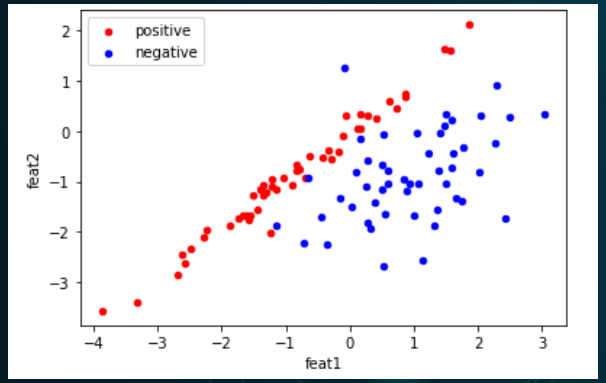
- Lựa chọn dữ liệu
- Nguy cơ data leakeage

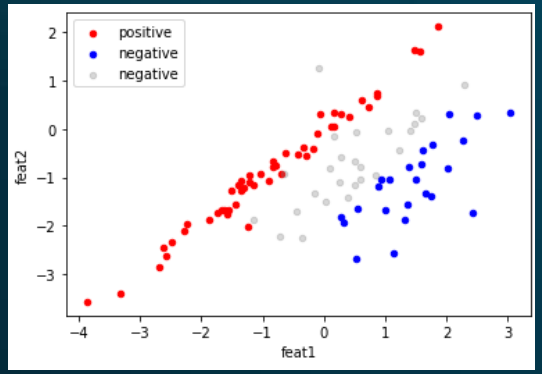
Lựa chọn nhãn negative

★ Bài toán thuộc loại binary classification, có 2 nhãn là positive và negative.

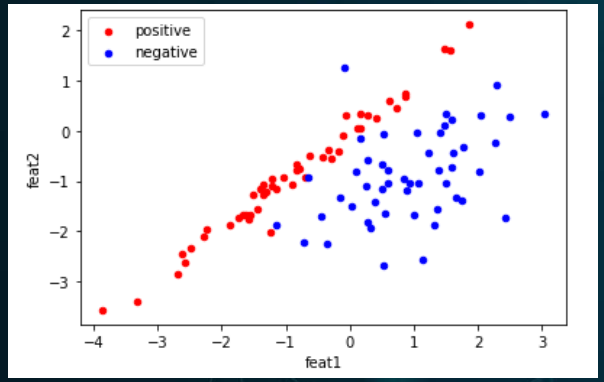
- Chọn an toàn: Chỉ sample trên tập có khả năng negative cao
- Chọn random: sample trên tập mass sau khi loại bỏ các positive sample.

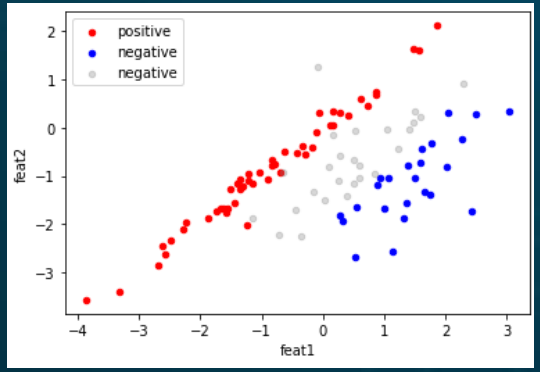
Lựa chọn nhãn negative

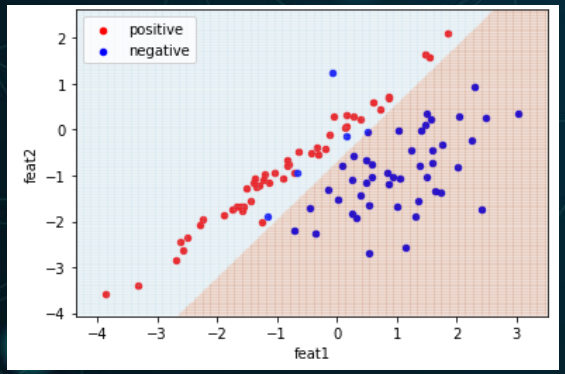


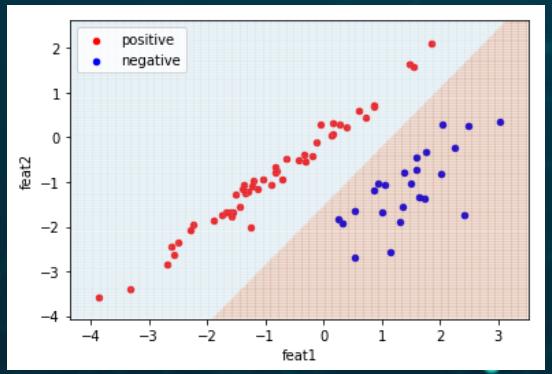


Lựa chọn nhãn negative

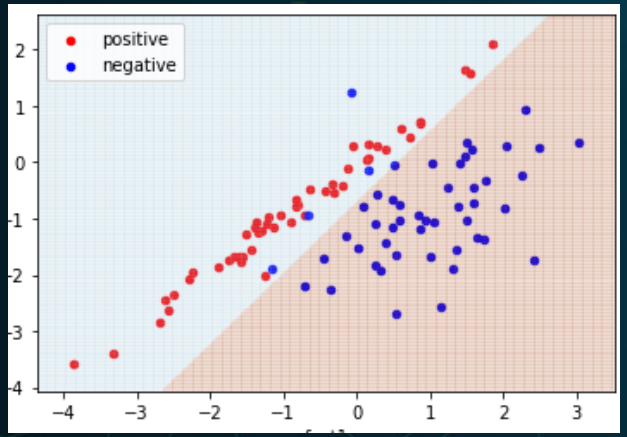


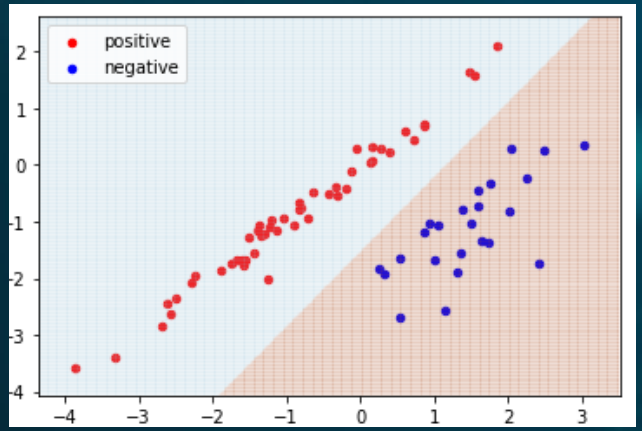


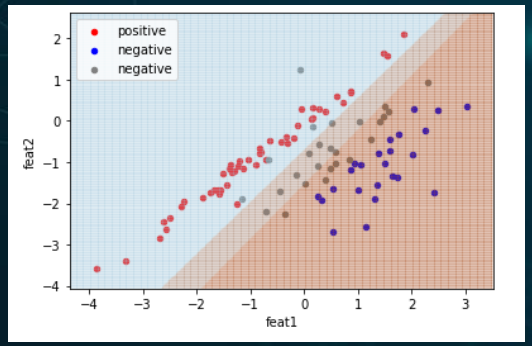




Lựa chọn nhãn negative







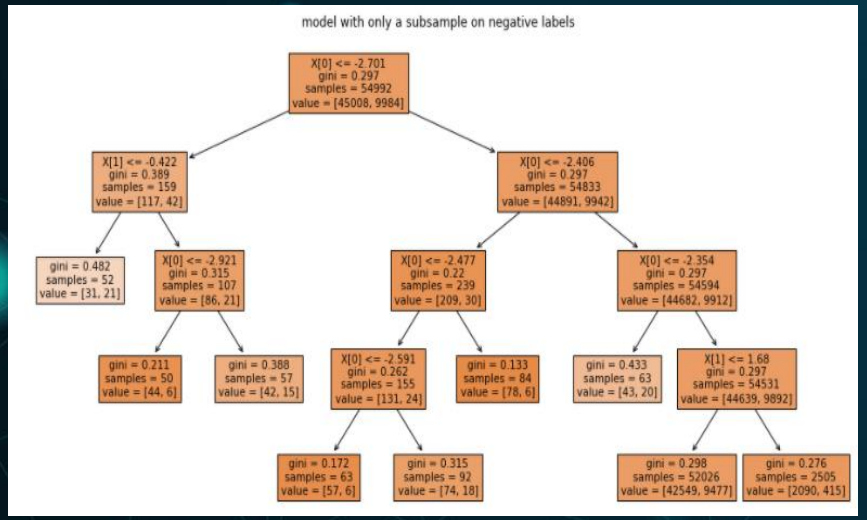
Lựa chọn nhãn negative

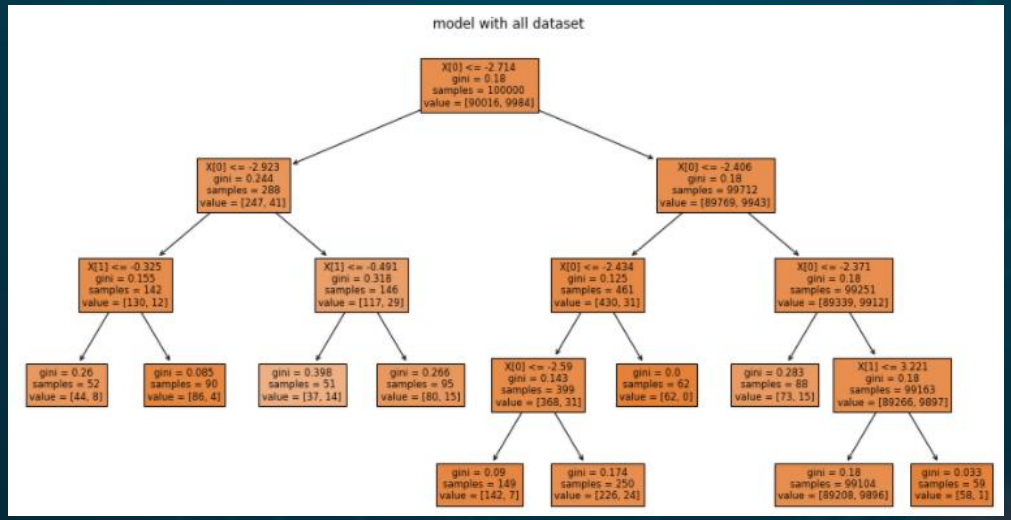
★ Bài toán thuộc loại binary classification, có 2 nhãn là positive và negative.

Tình huống 1: Với 1 mô hình cho trước: Có 3 tập là **A**: nhãn positive. Và **B, C** nhãn negative. Trên tập nào sẽ cho gini cao hơn? **A** U **B** hay **A** U **B** U **C** ?

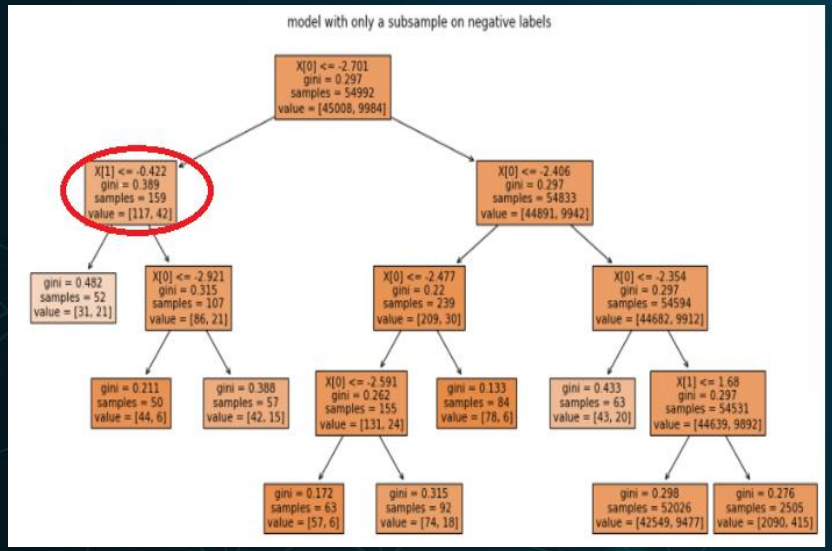
Tình huống 2: **A**: nhãn positive. Và **B, C** nhãn negative. Nên chọn bộ nào để training model? **A** U **B** hay **A** U **B** U **C** ?

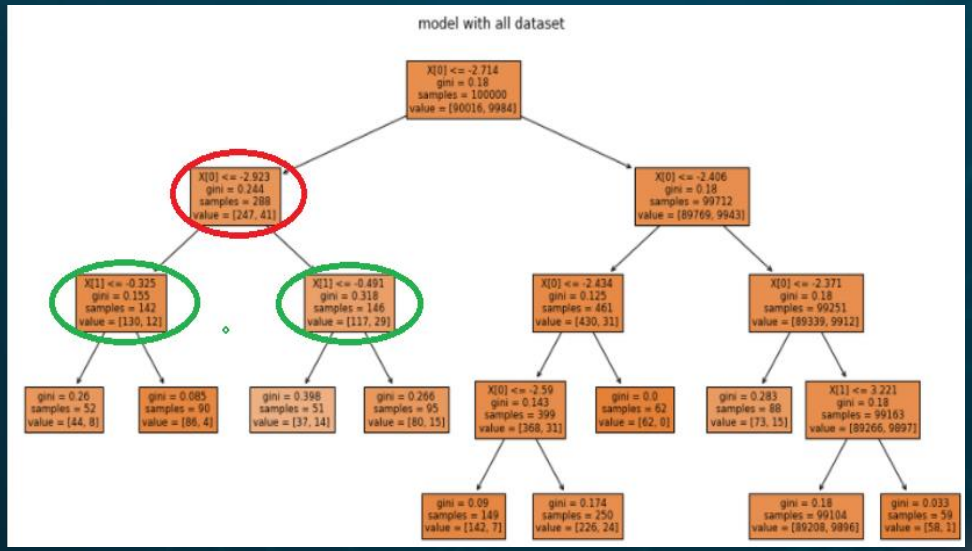
Lựa chọn nhãn negative





Lựa chọn nhãn negative





Lựa chọn nhãn negative

★ Bài toán thuộc loại binary classification, có 2 nhãn là positive và negative.

- IMPORTANCE: Dữ liệu trên tập train có cùng phân bố với dữ liệu trên toàn dataset
- Việc FILTTERING dữ liệu có thể có các nguy cơ làm cho dữ liệu không đại diện

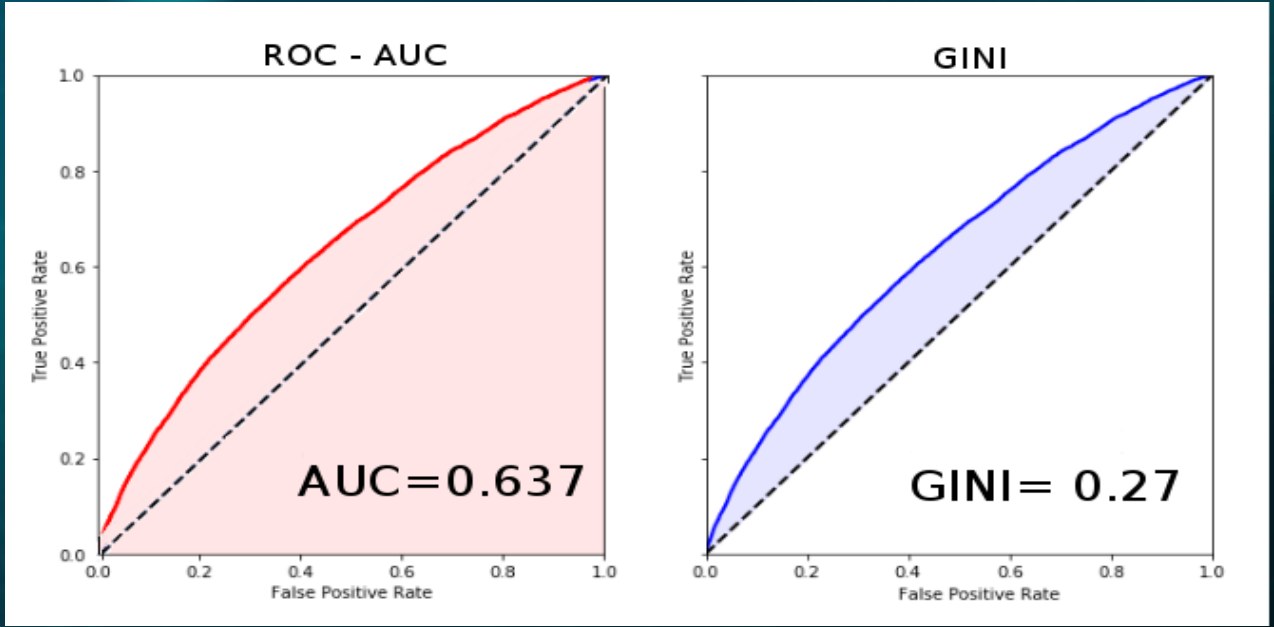
2

Metrics

- ★ Lựa chọn metric
- ★ Các đặc tính của AUC, GINI

Độ đo

Với bàn toán binary classification, metric thường dùng là AUC hay GINI



Các đặc tính của AUC - GINI

01 *Gini = 2* AUC -1*

02 *Cả 2 metrics này đều bất biến với các phép biến đổi đơn điệu*

03 *Ý nghĩa: Là 1 cách tính trung bình về tất cả các stratery về việc chọn ngưỡng cho 1 model về binary classification*

Câu hỏi

01 *GINI, AUC đã đặc trưng có phải là metrics tốt nhất?*

02 *Trong trường hợp nhãn rất lệch: số lượng positive << số lượng negative, các metrics trên còn hoạt động tốt?*

03 *AUC tốt hơn có chắc chắn thu được precision, recall tốt hơn?*

Features Extractions

3

- From transactions to features
- From features to features

From transaction to features

01 *Liệt kê các đặc tính của các transaction*

From transaction to features

01 *Liệt kê các đặc tính của các transaction*

02 *Kiểu numeric: sum, median, mean, std, entropy, percentiles, ...*

From transaction to features

01 *Liệt kê các đặc tính của các transaction*

02 *Kiểu numeric: sum, median, mean, std, entropy, percentiles, …*

03 *Kiểu categorical: mode, count distinct, count, group theo nhóm, hashing, …*

From transaction to features

01 *Liệt kê các đặc tính của các transaction*

02 *Kiểu numeric: sum, median, mean, std, entropy, percentiles, …*

03 *Kiểu categorical: mode, count distinct, count, group theo nhóm, hashing, …*

04 *Deal with missing: Count, add extrat feature is_missing, etc*

From transaction to features

01 *Liệt kê các đặc tính của các transaction*

02 *Kiểu numeric: sum, median, mean, std, entropy, percentiles, ...*

03 *Kiểu categorical: mode, count distinct, count, group theo nhóm, hashing, ...*

04 *Deal with missing: Count, add extra feature is_missing, etc*

05 *Deal with time series: Select time window, start date, end date*

From features to features

01 CAT: *Onehot encoder, label encoder, target encoder, gom nhóm,...*

02 NUMERIC: *Boxcox, disretization*

03 *Group of NUMERIC feature: map to tổng, hiệu, tích, thương.*

04 *Deal with missing data*

05 *Sử dụng meta data*

From features to features
How to generate relevant feature?

01 | *By model*

02 | *By tool*

03 | *Add extra features*

Modelling

4

- ★ Baseline model
- ★ Model selection
- ★ Hyper parameters tuning

Model Selection

01 *Sử dụng các model đơn giản như LogisticRegression hay SVM, kNN để tạo ra kết quả baseline*

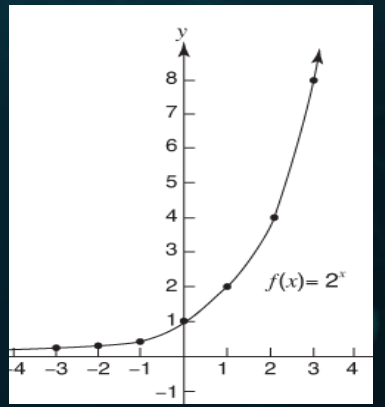
02 *Về lý thuyết, các mô hình Neural Netwwork có thể học được mọi cấu trúc dữ liệu và perform tốt trên mọi dữ liệu. Tuy nhiên, rất khó tinh chỉnh và training do cấu trúc phức tạp. Các mô hình boosting tree là sự lựa chọn tốt nhất.*

Hyper parameters tuning

01 *3 methods:* ***grid search, Random search, bayes search***

02 *Phương pháp* ***bayes search*** *thường cho kết quả tốt nhất.*

03 *Curse of dimensionality:* $d \rightarrow exp(d)$

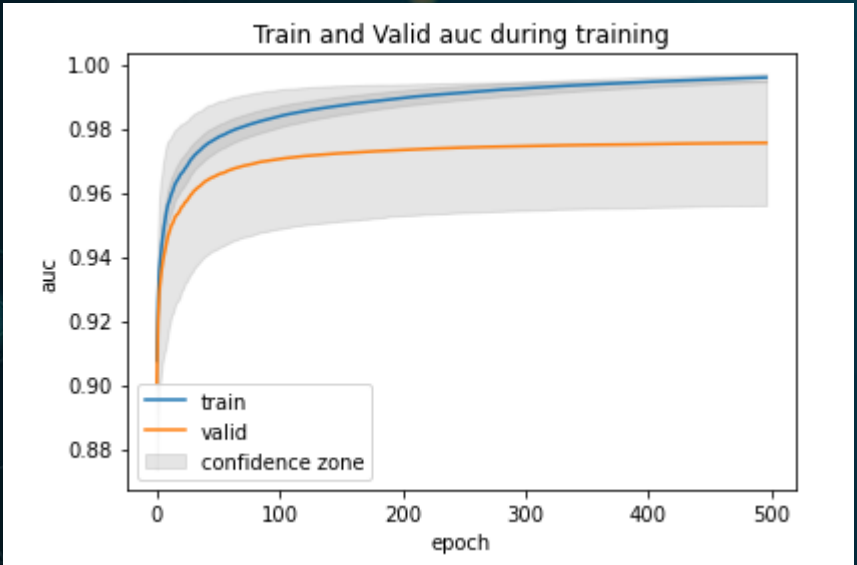


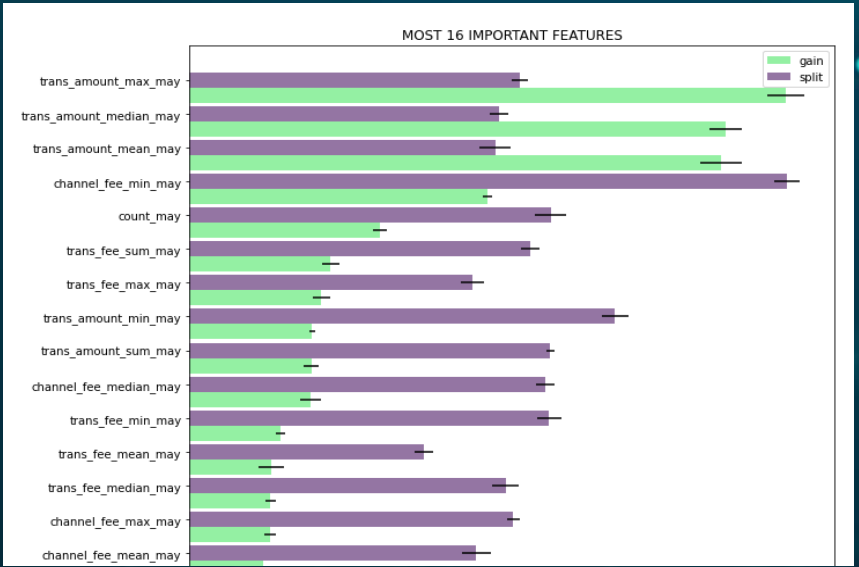
Model visualization

5

- ★ Basic visualization
- ★ Tree based visualization
- ★ Black box model visualization

Basic Visualization





Tree based visualization

- *Visualize Decision Tree*
- *Visualize path of decision tree*
- *Visualize attribute of features in decision tree*

```
pip install shap
```

https://github.com/slundberg/shap

https://suinlee.cs.washington.edu/

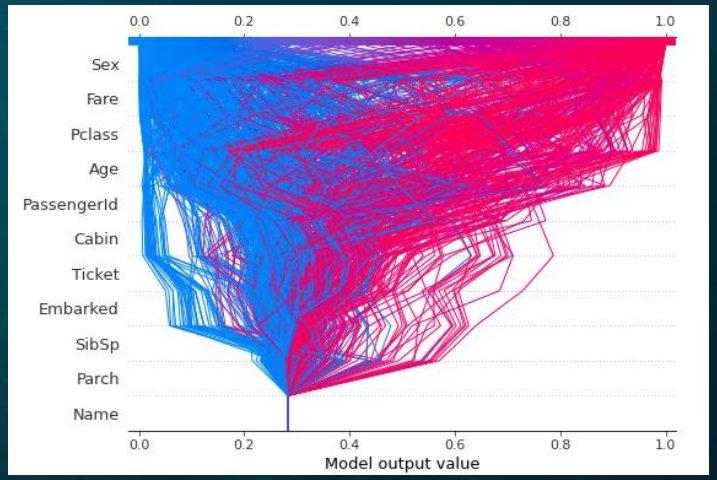
Tree based visualization

- *Visualize Decision Tree*
- *Visualize path of decision tree*
- *Visualize attribute of features in decision tree*

Visualize feature importance of a single output.
Feature list: PTRAIO, LSTAT, RM, NOX, AGE, RAD

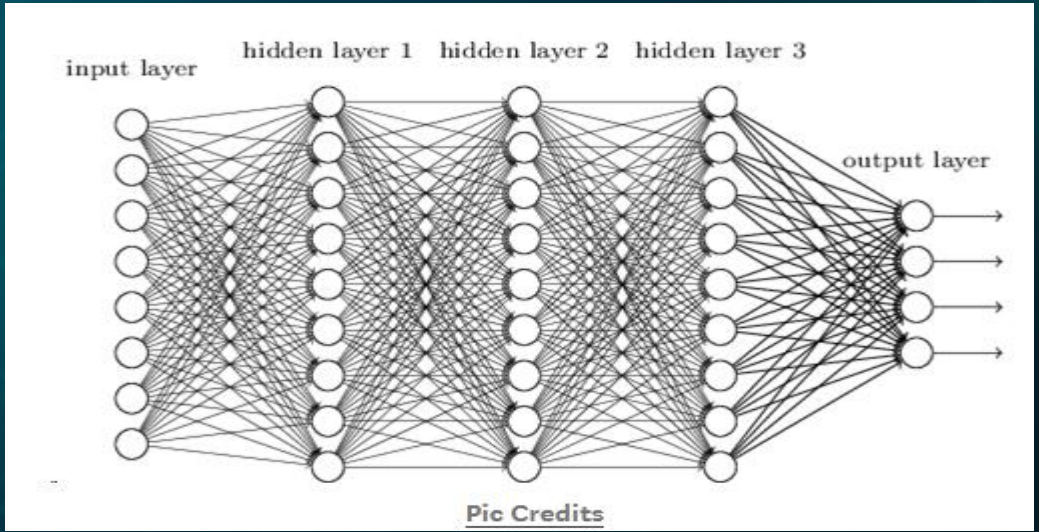
Tree based visualization

- *Visualize Decision Tree*
- *Visualize path of decision tree*
- *Visualize attribute of features in decision tree*



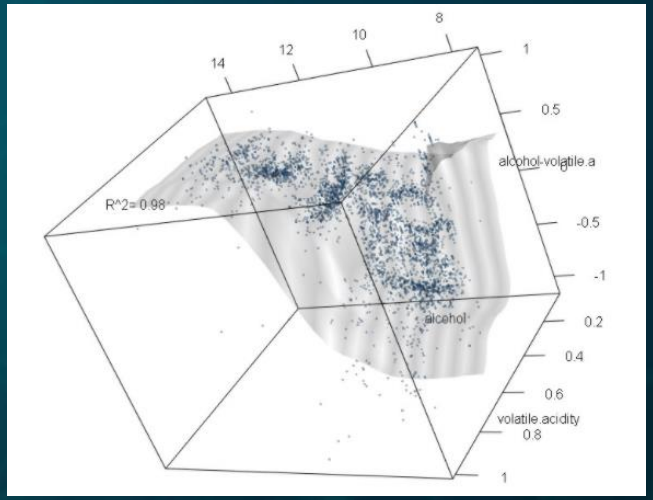
Black box model visualization

- No *direct visualization*
- *Can we estimate important feature for each output?*



Black box model visualization

- *No direct visualization*
- *Can we estimate important feature for each output?*



Black box model visualization

- No *direct visualization*
- *Can we estimate important feature for each output?*

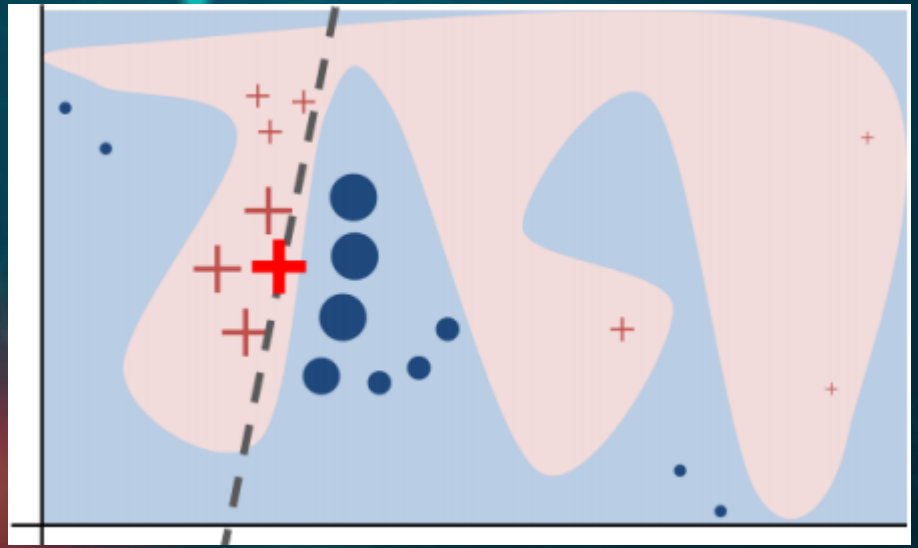
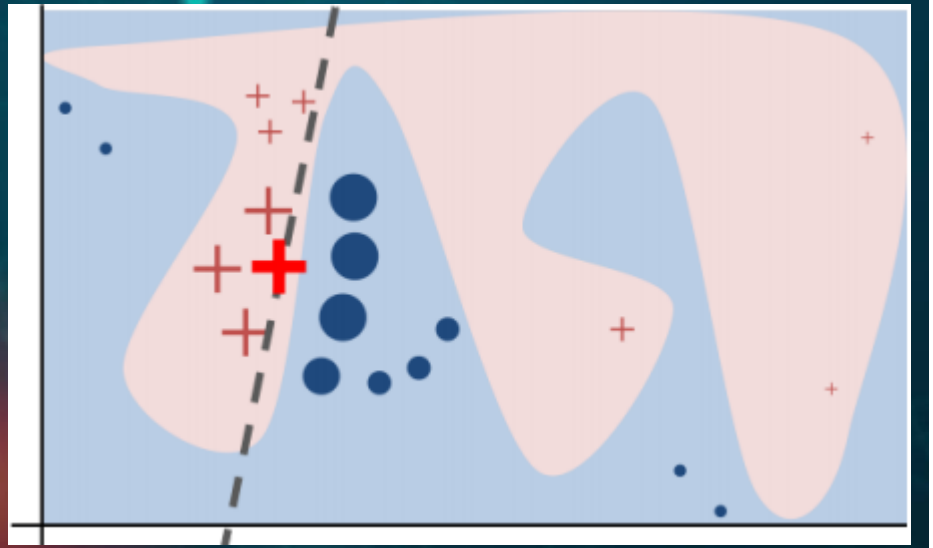


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

Black box model visualization

- No *direct visualization*
- *Can we estimate important feature for each output?*



Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, 3, ..., N\}$ **do**
    $z'_i \leftarrow sample\_around(x')$
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
  end for
  $w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
  return w

Black box model visualization

(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" ($p = 0.21$)

THANK YOU