Detecting discrimination through Suppes-Bayes Causal Network

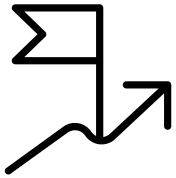
A bachelor's thesis by Blai Ras in collaboration with Eurecat

UNIVERSITAT DE BARCELONA

eurecat

Project

- Based on “Exposing the probabilistic causal structure of discrimination” by Francesco Bonchi, Sara Hajian, Bud Mishra & Daniele Ramazzotti.
- Started 21th October.
- Working part-time since 17th November.
- Delivery date: 19th January.

Goals 

1. Porting the algorithm to Python.

2. Extending & upgrading the algorithm once in Python.

3. Design & deploy a user-friendly website able to run the algorithm and show its results.

Discrimination Types

Group

Individual

Favoritism

Conditional Explainable

Data input

1. Dataset.

Admit_Admitted	Admit_Rejected	Gender_Female	Gender_Male	Dept_A	Dept_B	Dept_C	Dept_D	Dept_E
1	0	0	1	1	0	0	0	0
0	1	1	0	0	1	0	0	0
1	0	0	1	0	0	1	0	0
0	1	0	1	0	0	0	1	0
1	0	1	0	0	0	0	0	1

2. Temporal Order Table

Attribute	Order
Admit_Admitted	3
Admit_Rejected	3
Gender_Female	1
Gender_Male	1
Dept_A	2
Dept_B	2
Dept_C	2
Dept_D	2
Dept_E	2

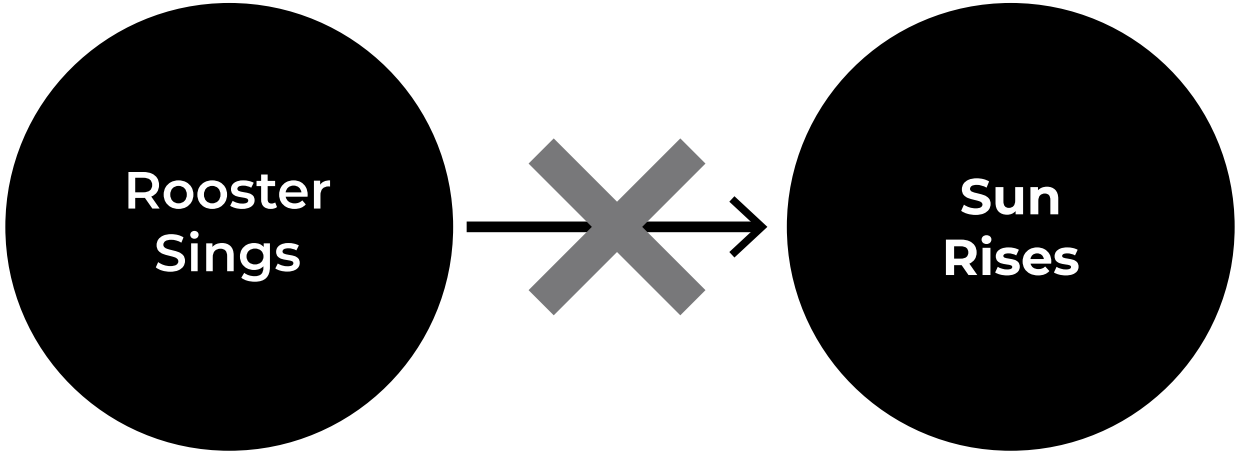
Suppes-Bayes Causal Network

Consists of:

1. Ensuring Suppes' conditions.
2. Training by Likelihood Fit.

Suppes' Conditions

a. Temporal Priority

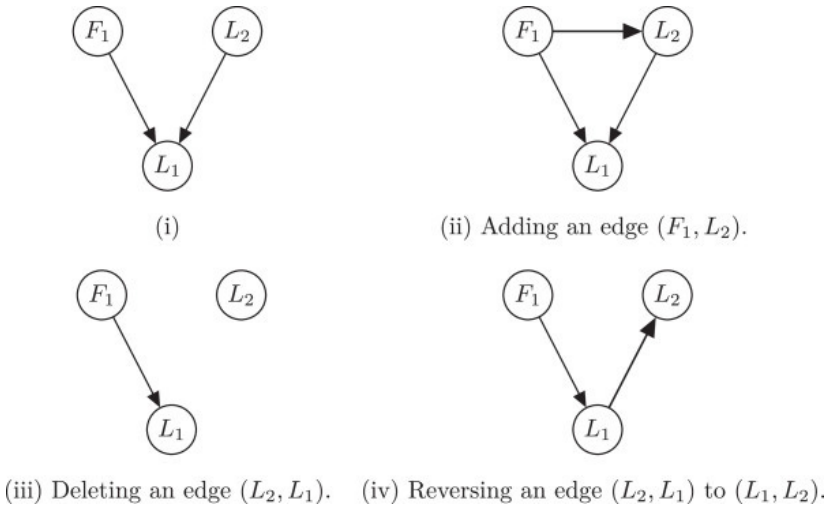


b. Probability Rising

$$P(v \mid u) > P(v \mid \neg u)$$

Training

- Hill Climbing algorithm.
- Bayesian Information Criteria.
- Logarithmic Likelihood function.

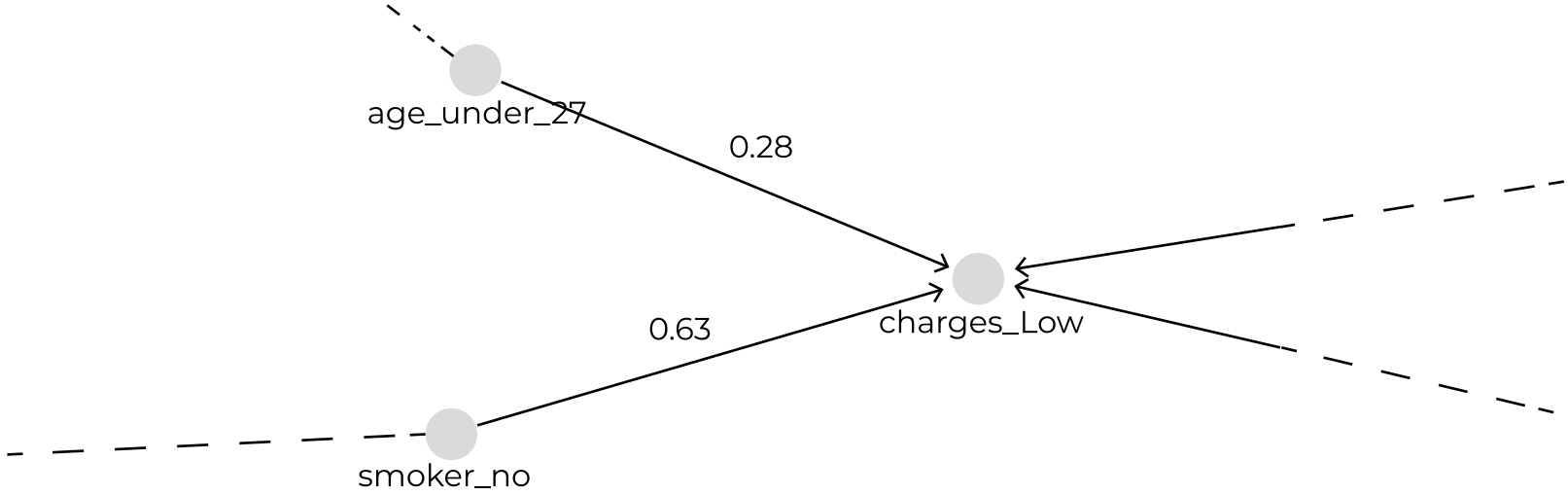


(i) (ii) Adding an edge (F_1, L_2).

(iii) Deleting an edge (L_2, L_1). (iv) Reversing an edge (L_2, L_1) to (L_1, L_2).

$$BIC = -2L_{log}(x) + k\ log(n)$$

Final Network



$$P(u \mid v) - P(u \mid \neg v)$$

Discrimination measurement

Weighted Random Walk

$$\frac{\text{walks leading to positive or negative decisions}}{\text{total walks}}$$

But what if walker ends on a leaf node...?

Discrimination measurement

Weighted Random Walk

Total Inconclusive Score

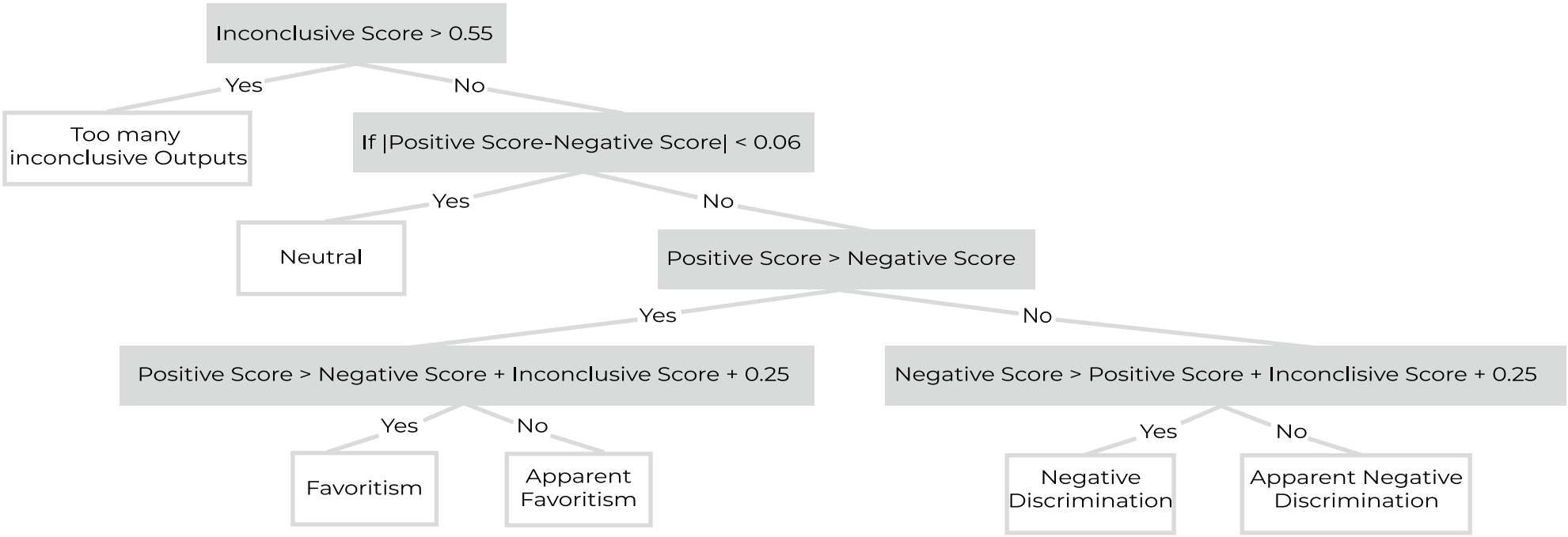
- Number of inconclusive walks > 0,55.

Partial Inconclusive Score

- Apparent positive or negative discrimination.
- Difference between positive or negative scores and inconclusive score < 0,25.

Discrimination measurement

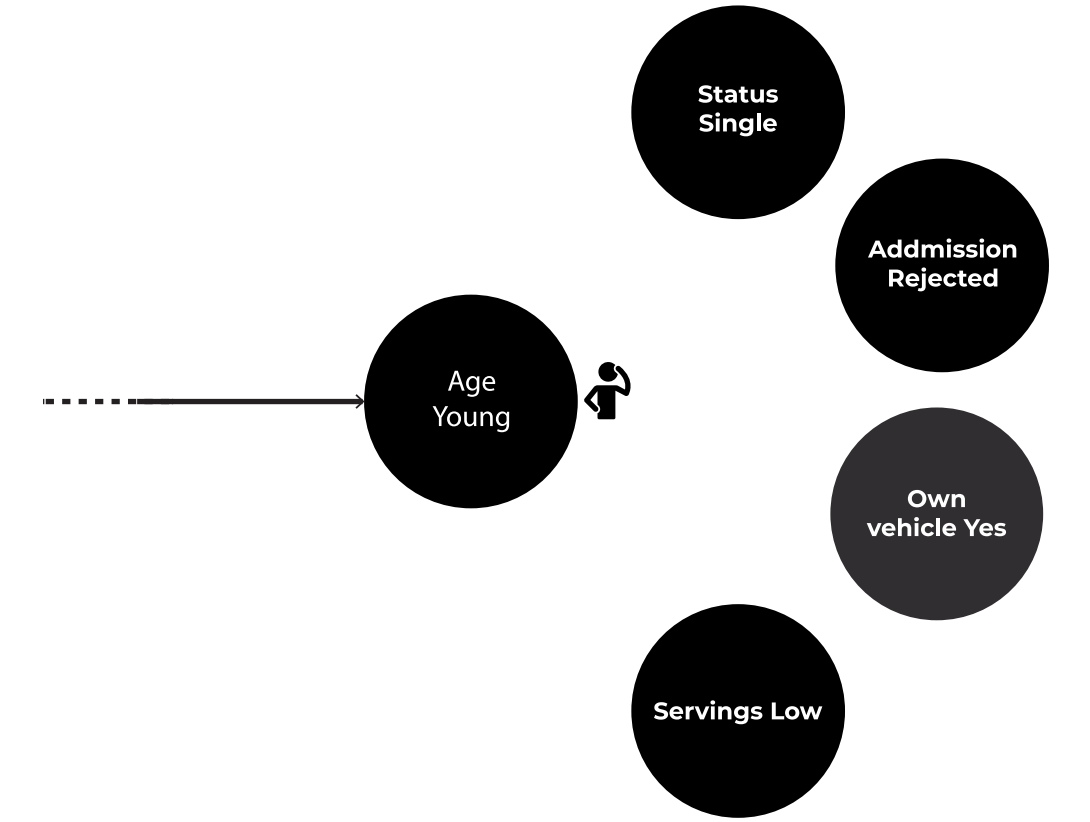
Veredicts



Discrimination measurement

- **Personalized Page Rank**

  - Once on a leaf node, the probability of jumping to another node is given by the individual attributes.



Savings Low	Own vehicle Yes	Admission Rejected	Status Single
1	0	1	1

Discrimination measurement

- **Personalized Page Rank**

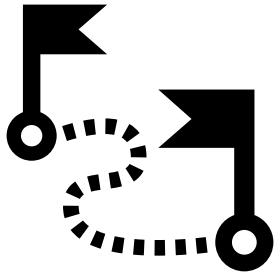
  - Every node has a score, but we are only interested in the positive/negative decision nodes.

Website

- **Developed in Python with Django framework**
- **Backend**
  - Multipart/Form Data with the datasets saved in a SQLite Database.
  - Algorithm.
- **Frontend**
  - CSS + Bootstrap.
  - JavaScript.
- **Deploy**
  - AWS Cloud server running Ubuntu 18.04.

Experiments

http://ec2-34-225-210-97.compute-1.amazonaws.com:8000/



Conclusions

- Algorithm successfully ported to Python.
- Algorithm upgraded and extended with inconclusive score.
- Website up & running, able to run the algorithm and show its results in a visual and easy to understand way.

Future Work

- Use the *bnlearn* Python library.
- Remove the Temporal Order table.
- Find the best discrimination thresholds depending on the data.

Thank you!

Blai Ras