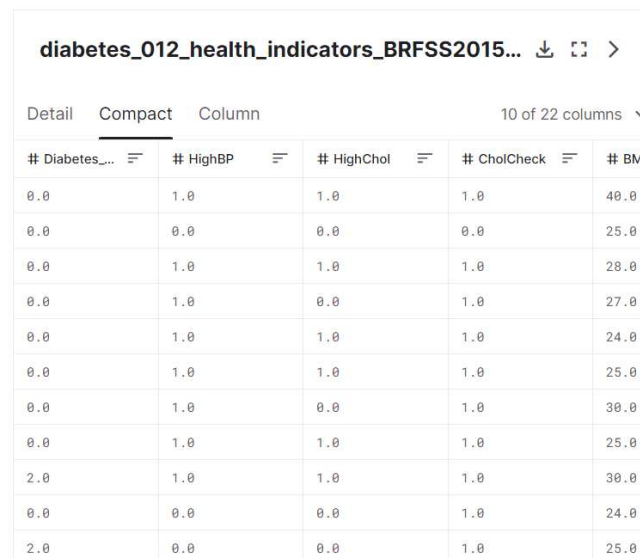


1. Dataset

Dataset yang digunakan:
https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_012_health_indicators_BRFSS2015.csv



| # Diabetes... | # HighBP | # HighChol | # CholCheck | # BM |
|---------------|----------|------------|-------------|------|
| 0.0 | 1.0 | 1.0 | 1.0 | 40.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 25.0 |
| 0.0 | 1.0 | 1.0 | 1.0 | 28.0 |
| 0.0 | 1.0 | 0.0 | 1.0 | 27.0 |
| 0.0 | 1.0 | 1.0 | 1.0 | 24.0 |
| 0.0 | 1.0 | 1.0 | 1.0 | 25.0 |
| 0.0 | 1.0 | 0.0 | 1.0 | 30.0 |
| 0.0 | 1.0 | 1.0 | 1.0 | 25.0 |
| 2.0 | 1.0 | 1.0 | 1.0 | 30.0 |
| 0.0 | 0.0 | 0.0 | 1.0 | 24.0 |
| 2.0 | 0.0 | 0.0 | 1.0 | 25.0 |

Dataset ini digunakan karena lengkap, cocok untuk pengkategorian RDR karena kebanyakan kolom sudah berupa Boolean. Data juga sangat banyak mencapai 200000 lebih row dan label lengkap. Data juga tidak ada missing value signifikan. Data yang digunakan adalah data dengan tiga label agar lebih menarik diagnosisnya. Kemudian kolom non-boolean semua di-drop dan dataset dibagi menjadi latih dan uji.

2. Implementasi

RDR Diimplementasikan dalam kelas Python meniru cara kerja Model pada pustaka Scikit-Learn, karena menurut saya model mereka mudah digunakan dan saya ingin mengemulasi hal tersebut dalam kode saya.

Ada tiga fungsi utama yang diimplementasikan, yaitu:

1. Fit

Menggunakan Dataset dan label untuk melatih model RDR Dalam implementasi, hal ini dilakukan dengan mengambil pasangan fitur dan label yang sudah diolah menjadi list dan dimasukkan ke dalam pohon RDR menggunakan aturan RDR yang ada. Beberapa term yang digunakan:

- contradict, hal ini terjadi ketika terdapat konklusi, namun pakar tidak setuju sehingga diubah, bisa dengan memasukkan fitur sendiri (tidak dalam fit) atau dipilih secara acak dengan strategi

- conclusion leaf, hal ini adalah daun pohon RDR yang mencapai konklusi yang dikontradiksi oleh pakar atau label data
- manifest, hal ini terjadi ketika mendapatkan konklusi yang tidak sesuai dan di daun selanjutnya, fitur tidak memenuhi prasyarat konklusi. Maka rule dapat dikatakan gagal manifestasi sehingga dinamakan manifestasi.

2. Predict

Predict digunakan untuk memprediksi kategori dataset yang dimasukkan ke dalam model, hasil akan dikeluarkan sebagai suatu daftar hasil

3. Add_New_Cornerstone

Fungsi ini khusus untuk masukan pakar sehingga bisa mengatur konklusi dan prekondisi apa saja yang ingin ditentukan oleh rule tersebut, kegunaan dicontohkan di bagian bawah

Detail implementasi lengkap lebih lengkap ada pada Jupyter Notebook yang disediakan pada laman: <https://github.com/nart4hire/RDR>. Atau yang dikirim sebagai pdf di Edunex.

3. Hasil

Akurasi tidak bisa mencapai terlalu tinggi jika hanya menggunakan supervised learning (membiarkan mesin mempelajari dataset sendiri dengan label). Lebih baik digunakan oleh pakar dalam bidangnya. Detil lebih lanjut ada pada Jupyter Notebook.

4. Tantangan

Menerjemahkan Dataset menjadi daftar fitur merupakan tantangan besar karena tidak tersedia pustaka yang mudah untuk melakukannya.

5. Saran Pengembangan Selanjutnya

- Mengembangkan WebUI atau GUI untuk mempermudah masukan dari pakar yang kurang mengerti koding.
- Mengembangkan pemrosesan data internal agar input dengan berbagai format dan format yang tidak ideal tetap dapat digunakan.