

Handleiding HUP_Generator

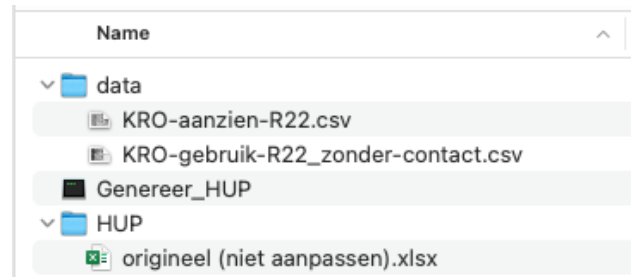
De HUP_Generator is een programma wat aan de hand van de KRO een nieuwe HUP genereert. De door de VRBZO opgestelde risico matrix welke is gecommuniceerd met de Gemeentes dicteert de filter criteria.

Toegang

De originele code is te vinden op Github via de volgende link: https://github.com/TJKNL/VRBZO_HUP_Generator. Onder het kopje *Releases* vind je alle versies van het programma.

Gebruik

Als het .exe bestand is gedownload dient het in een map met de volgende structuur geplaatst te worden. Namen van mappen en bestanden dienen overgenomen te worden zoals in de voorbeeld afbeelding rechts. Wanneer de .exe wordt aangeklikt komt er in het mapje "HUP" een nieuwe HUP in Excel formaat.

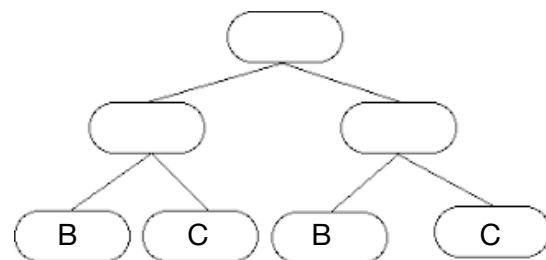


Werking

Hieronder wordt de logica van de HUP_Generator uitgelegd aan de hand van de componenten.

De Boom

Het programma werkt via een boom structuur. Objecten gaan van boven naar beneden wanneer ze aan een filter criteria voldoen. Voldoen ze niet aan een filter voldoen, gaan ze niet verder naar beneden in de boom. Enkel de onderste bladeren van de boom krijgen een risico B of C classificatie.



Filters

De eerder genoemde filters definiëren de criteria (zie code regel 40 en 41). Een filter bestaat uit drie onderdelen, namelijk: een KRO kolom, een vergelijking (bv: ==, >, <, ...) en een waarde. Voor binaire kolommen staat 0 voor *niet waar* en 1 voor *waar*.

```
39  ###
40  tree.filter("winkelfunctie", "==", 1)
41  tree.filter("woz_opp_nietwoon", ">", 1000)
42  # TODO: Wonen boven winkels data is niet beschikbaar of uit de KRO te halen.
43  tree.set_risk("C")
44  tree.store_results()
45  tree.reset()
```

Risico Classificatie

Het 'tree' object behoudt 1 huidig pad van boven naar beneden in de boom. Wanneer het risico wordt geselecteerd (zie code regel 43), gebeurt dit voor alle objecten op dat moment nog in de boom. Wanneer de boom wordt gereset (zie code regel 45), ben je weer terug boven aan de boom samen met alle objecten die origineel in de boom zaten. Dus ook alle reeds geclassificeerde objecten.

***Voor de exacte werking, de code is voorzien van uitgebreide uitleg
(Zie: main.py, classes.py & data_management.py)***