**APPENDIX A – Genetic Algorithm**

# 1 – Introduction

The purpose of this appendix is ​​to explain the genetic algorithm created to improve the parameterization of the DecisionTreeClassifier.

After not getting good results with filling the NaN values with basic machine learning algorithms, we decided to create a genetic algorithm for tuning the decision three algorithm for better results.

We’ve made tests on all the numerical columns that have NaNs, which are:

- Gross Monthly Salary (Index 10)

- Premiums in LOB: Motor (Index 13)

- Premiums in LOB: Health (Index 15)

- Premiums in LOB: Life (Index 16)

- Premiums in LOB: Work Compensations (Index 17)

- Age (Index 18)

- First Policy´s Age (Index 19)

# 2 – Rules

* + We ran 10 times the algorithm for each column;
  + We only used results from columns with R Squared above 0.60;
  + To measure the fitness, we split the complete dataset intro train and test (65% and 35%), and after applying the regressor we measured the R Squared using predictions x true values from test set;
  + The population size is 30 and the number of generations for each run is 100;
  + The Mutation-Probability is 0.5 and the Crossover-Probability is 0.8;
  + We used single point crossover, single point mutation and roulette wheel selection;
  + The code for the GA algorithm is at GitHub at the following link: <https://github.com/apanchot/data_mining/tree/master/GA%20for%20ML> .
* The constraints were:

DT\_constraints = {

    "min\_sample\_split" : [0,301],

    "min\_samples\_leaf"“:[0,301],

    "max\_features": [0,10],

    "max\_depth" : [0,30]

}

The representation it’s an array of length 5 which:

The first element represents the criterion parameter.

If 0 == ‘mse’;

If 1 == ‘friedman\_mse'

If 2 == ‘mae’

The second element represents the min\_sample\_split parameter;

The third element represents the min\_samples\_leaf parameter;

The fourth element represents the max\_features parameter;

The fifth element represent the max\_depth parameter.

Example:

[1, 23, 10, 9, 8] = [criterion=’friedman\_mse’, min\_sample\_split=23, min\_samples\_leaf=10, max\_features=9, max\_depth=8]

**Tests**

**Column index 10 – Gross Monthly Salary**



The best solution was Test 6 - [1, 60, 133, 9, 16] - fitness: 0.4788 – R² = 47.88%

[criterion= ‘friedman\_mse’,

min\_sample\_split=60,

min\_samples\_leaf=133,

max\_features=9,

max\_depth=16]

This solution is less than 0.5, we will **discard** it and not using Decision Three on Gross monthly salary.

**Column index 13 - Premiums in LOB: Motor**

The best solution was Test 5 - [1, 122, 1,3, 28] - fitness: 0.8157 – R² = 81.57%

[criterion= ‘friedman\_mse’,

min\_sample\_split=81,

min\_samples\_leaf=7,

max\_features=9,

max\_depth=26]

This solution is more than 0.5, we **use this solution** for predicting Premiums in LOB: Motor null values.

**Column index 15 - Premiums in LOB: Health**



The best solution was Test 6 - [0, 42, 5, 9, 14] - fitness: 0.1316 – R² = 13.16%

[criterion= ‘mse’,

min\_sample\_split=42,

min\_samples\_leaf=5,

max\_features=9,

max\_depth=14]

This solution is less than 0.5, we will **discard** it and not using Decision Three on Premiums in LOB: Health.

**Column index 16 - Premiums in LOB: Life**



The best solution was Test 4 - [1, 2, 3, 9, 9] - fitness: 0.6286 – R² = 62.86%

[criterion= ‘friedman\_mse’,

min\_sample\_split=2,

min\_samples\_leaf=3,

max\_features=9,

max\_depth=9]

This solution is more than 0.5, we **use this solution** for predicting Premiums in LOB: Life null values.

**Column index 17 - Premiums in LOB: Work Compensations**



The best solution was Test 9 - [1, 52, 8, 7, 25] - fitness: 0.4089 – R² = 40.89%

[criterion= ‘friedman\_mse’,

min\_sample\_split=52,

min\_samples\_leaf=8,

max\_features=7,

max\_depth=25]

This solution is less than 0.5, we will **discard** it and not using Decision Three on Premiums in LOB: Work Compensations.

**Column index 18 – Age**



The best solution was Test 1 - [0, 8, 36, 9, 13] - fitness: 0.8817 – R² = 88.17%

[criterion= ‘mse’,

min\_sample\_split=8,

min\_samples\_leaf=36,

max\_features=9,

max\_depth=13]

This solution is more than 0.5, we **use this solution** for predicting Premiums Age values.

PS.: After talking to teacher Fernando Bação we discovered that the Age columns doesn’t look right, we decided to drop it.

**Column index 19 – First Policy´s Age**



The best solution was Test 6 - [1, 106, 38, 2, 1] - fitness: 0.0006 – R² = 00.06%

[criterion= ‘friedman\_mse’,

min\_sample\_split=106,

min\_samples\_leaf=38,

max\_features=2,

max\_depth=1]

This solution is less than 0.5, we will **discard** it and not using Decision Three on First Policy´s Age.

Summarize:

- Gross Monthly Salary (Index 10)   
 Discarded

- Premiums in LOB: Motor (Index 13)

The best solution was Test 5 - [1, 122, 1,3, 28] - fitness: 0.8157 – R² = 81.57%

- Premiums in LOB: Health (Index 15)

Discarded

- Premiums in LOB: Life (Index 16)

The best solution was Test 4 - [1, 2, 3, 9, 9] - fitness: 0.6286 – R² = 62.86%

- Premiums in LOB: Work Compensations (Index 17)

Discarded

- Age (Index 18)

The best solution was Test 1 - [0, 8, 36, 9, 13] - fitness: 0.8817 – R² = 88.17%

- First Policy´s Age (Index 19)

Discarded