Introduction

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 will make testes on all the numerical columns that have NaN's, 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)

Rules

We will run 10 times the algorithm for each column.

We will only use for 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 will be on our GitHub.

The constraints was:

```
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

Column index 10 - Gross Monthly Salary		
ID	REP	FITNESS
TEST 1	[0, 243, 32, 9, 23]	0,477276686
TEST 2	[1, 6, 115, 9, 11]	0,477709166
TEST 3	[0, 230, 84, 7, 26]	0,473805739
TEST 4	[0, 219, 106, 9, 28]	0,478128313
TEST 5	[0, 246, 55, 7, 20]	0,475999419
TEST 6	[1, 60, 133, 9, 16]	0,478825235
TEST 7	[1, 17, 177, 7, 6]	0,473022784
TEST 8	[0, 91, 126, 9, 16]	0,478801144
TEST 9	[1, 99, 60, 9, 27]	0,478328645
TEST 10	[1, 218, 49, 8, 15]	0,475193149

The best solution was Test 6 - [1, 60, 133, 9, 16] - fitness: 0.4788 – R^2 = 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

Column index 13 - Premiums in LOB: Motor		
ID	REP	FITNESS
TEST 1	[1, 81, 7, 9, 26]	0,531480899
TEST 2	[0, 10, 9, 7, 9]	0,476222785
TEST 3	[0, 142, 7, 8, 20]	0,523394465
TEST 4	[1, 256, 7, 9, 23]	0,518537997
TEST 5	[1, 122, 1, 3, 28]	0,815182243
TEST 6	[1, 156, 22, 9, 23]	0,289806812
TEST 7	[1, 144, 20, 9, 9]	0,293341778
TEST 8	[1, 183, 7, 8, 19]	0,522199203
TEST 9	[0, 87, 44, 8, 17]	0,271270284
TEST 10	[1, 220, 4, 8, 12]	0,515576511

The best solution was Test 5 - [1, 122, 1,3, 28] - fitness: 0.8157 – R^2 = 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

Column index 15 - Premiums in LOB: Health		
ID	**REP**	**FITNESS**
TEST 1	[0, 46, 13, 9, 25]	0,056393847
TEST 2	[0, 31, 7, 4, 15]	0,094696699
TEST 3	[1, 6, 8, 9, 20]	0,086093738
TEST 4	[1, 121, 56, 5, 12]	0,018144267
TEST 5	[0, 126, 12, 7, 17]	0,057738533
TEST 6	[0, 42, 5, 9, 14]	0,131639069
TEST 7	[1, 164, 21, 4, 8]	0,037145065
TEST 8	[0, 19, 29, 6, 14]	0,016580034
TEST 9	[0, 248, 22, 5, 19]	0,035134211
TEST 10	[1, 191, 4, 4, 15]	0,155269953

The best solution was Test 6 - [0, 42, 5, 9, 14] - fitness: 0.1316 – R^2 = 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

Column 16 - Premiums in LOB: Life		
ID	REPRESENTATION	FITNESS
TEST 1	[1, 46 ,10, 9, 28]	0.6201194499604908
TEST 2	[0, 82, 32, 9, 26]	0.5921531472303051
TEST 3	[0, 79, 67, 9, 11]	0.5645884754229019
TEST 4	[1, 2, 3, 9, 9]	0.6286779273936715
TEST 5	[1, 29, 24, 9, 13]	0.5845829375832305
TEST 6	[0, 44, 56, 9, 20]	0.5722083169781447
TEST 7	[1, 5, 21, 8, 19]	0.5766313476368804
TEST 8	[0, 121, 26, 9, 23]	0.5768651510799641
TEST 9	[1, 41, 15, 9, 22]	0.6079690901162504
TEST 10	[1, 14, 12, 9, 9]	0.6020106651679107

The best solution was Test 4 - [1, 2, 3, 9, 9] - fitness: 0.6286 – R^2 = 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

Column index 17 - Premiums in LOB: Work Compensations		
ID	REP	FITNESS
TEST 1	[0, 5, 96, 8, 9]	0,331477986
TEST 2	[0, 33, 14, 9, 7]	0,405035868
TEST 3	[1, 74, 2, 5, 28]	0,408036522
TEST 4	[1, 88, 32, 9, 17]	0,37940321
TEST 5	[1, 9, 5, 8, 17]	0,394545105
TEST 6	[0, 68, 6, 8, 18]	0,384649996
TEST 7	[1, 86, 53, 9, 22]	0,377327598
TEST 8	[0, 146, 25, 9, 10]	0,368992978
TEST 9	[1, 52, 8, 7, 25]	0,408972296
TEST 10	[0, 53, 74, 9, 27]	0,353070222

The best solution was Test 9 - [1, 52, 8, 7, 25] - fitness: 0.4089 – R^2 = 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

Column index 18 – Age		
ID	REP	FITNESS
TEST 1	[0, 8, 36, 9, 13]	0,881717191
TEST 2	[1, 189, 76, 8, 29]	0,880112799
TEST 3	[1, 236, 87, 8, 16]	0,88034088
TEST 4	[1, 5, 66, 8, 7]	0,881111537
TEST 5	[0, 266, 11, 8, 27]	0,876731747
TEST 6	[1, 242, 34, 8, 12]	0,880314792
TEST 7	[0, 210, 120, 8, 17]	0,875409194
TEST 8	[1, 77, 113, 5, 20]	0,865051458
TEST 9	[0, 143, 94, 8, 29]	0,879270313
TEST 10	[1, 58, 120, 8, 29]	0,875409194

The best solution was Test 1 - [0, 8, 36, 9, 13] - fitness: 0.8817 – R^2 = 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

Column index 19 – First Policy´s Age		
ID	REP	FITNESS
TEST 1	[0, 22, 5, 1, 1]	-1,12523E-05
TEST 2	[1, 34, 226, 7, 2]	-0,000230829
TEST 3	[0, 254, 190, 1, 20]	-0,004908259
TEST 4	[1, 290, 162, 5, 1]	-0,000407931
TEST 5	[1, 146, 178, 5, 1]	0,000198636
TEST 6	[1, 106, 38, 2, 1]	0,000625661
TEST 7	[0, 198, 56, 2, 1]	0,000625661
TEST 8	[1, 225, 150, 4, 4]	-0,000737923
TEST 9	[1, 225, 251, 3, 2]	-0,000218549
TEST 10	[1, 99, 52, 1, 1]	-1,12523E-05

The best solution was Test 6 - [1, 106, 38, 2, 1] - fitness: 0.0006 – R^2 = 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 – R^2 = 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 – R^2 = 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 – R^2 = 88.17%

- First Policy´s Age (Index 19)

    Discarded