UNIT 7: KE5108
DEVELOPING INTELLIGENT SYSTEMS FOR PERFORMING BUSINESS ANALYTICS

OPTIMIZATION & FORECASTING

Workshop 1A & 1B: Online Advertising Plan
Workshop 2B: Direct Mailing Campaign for A Bank

SUBMISSION DATE: 30 SEPTEMBER 2018

TEAM GENESIS
SIDDHARTH PANDEY
PRANSHU RANJAN SINGH
NYON YAN ZHENG
TAN KOK KENG

MASTER OF TECHNOLOGY IN KNOWLEDGE ENGINEERING
KE30 (2018)

WORKSHOP 1A & 1B

Objective
To develop a hybrid intelligent system that can find an assignment of ad banners to websites and the start time and duration for each ad banner's display time which will maximise the user clicks while ensuring the budget requirement is met.

Tools
We have used the 2 tools; Microsoft Excel (Data Analysis and Solver) and Java library JGAP to solve this problem.

Data Description
The data consists of 1,000 observations of ad banner placement for each website. The start and end time, the total user clicks and cost for observation were given in the data.

We have calculated total duration for all the 5 websites for each day and computed the clicks per hour. A histogram of the clicks per hour is shown in Figure 1.

We can see that on average there are 9,748 clicks per hour. The maximum clicks per hour is 16,647. This information is later used as one of the constraints in the GA model in Workshop 1B to reduce the search space.

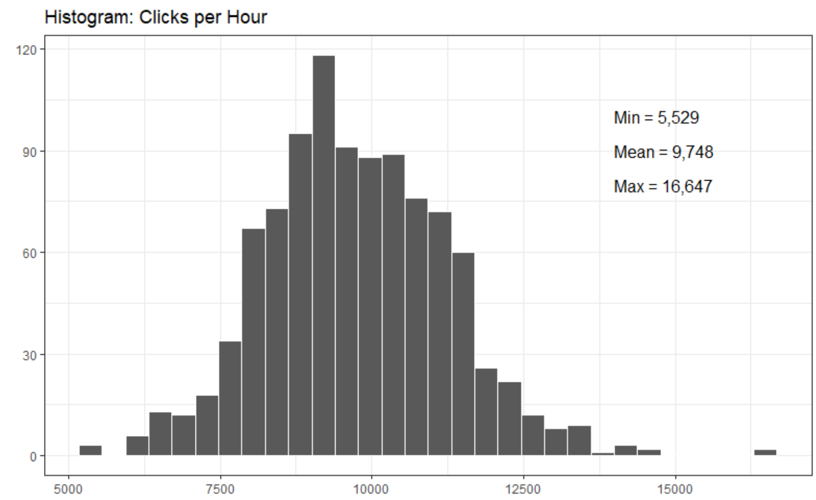


Figure 1 Histogram of Clicks per Hour

We have also computed the total ad placement duration by the website and by the banner, shown in Table 1. This information was later used as a guide for the initial values in the GA models.

Table 1 Total duration by Ad and Website

Ad	Website1	Website2	Website3	Website4	Website5	Total
1	621.3	603.4	649.4	884.5	844.9	**3,603.5**
2	571.9	813.6	1,154.7	502.7	719.5	**3,762.4**
3	475.8	821.8	682.7	823.1	744.7	**3,548.1**
4	694.3	577.7	891.9	824.0	863.7	**3,851.6**
5	529.2	843.6	803.6	815.6	584.2	**3,576.2**
6	620.4	822.8	833.3	688.4	707.3	**3,672.2**
Total	**3,512.9**	**4,482.9**	**5,015.6**	**4,538.3**	**4,464.3**	**22,014.0**

WORKSHOP 1A

Architecture

The architecture of the hybrid intelligent system is shown in Figure 2.

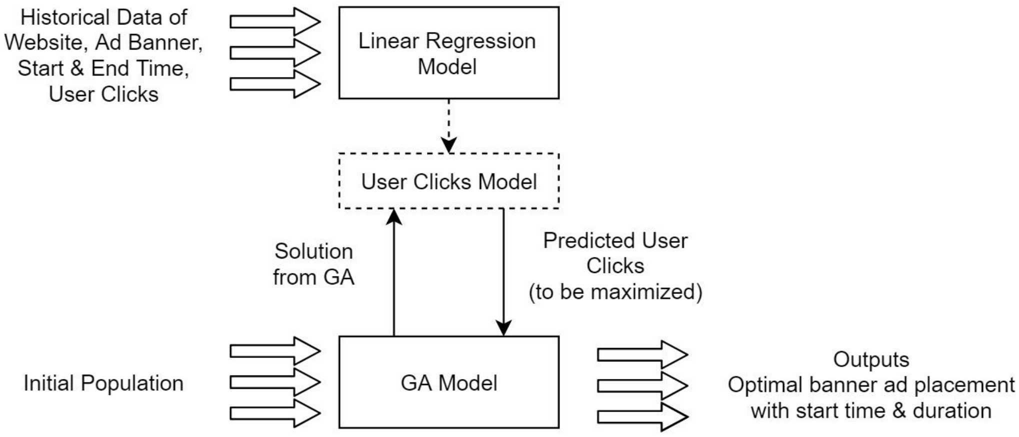


Figure 2 Workshop 1A Architecture

Linear Regression Model: To Predict User Clicks

A linear regression model is fitted to all the 1,000 observations and the resulting model is used to predict the number of user clicks given the website, ad banner, the start and end time.

The formula used to fit the regression model is

$$User\ Click \sim Start\ time_w + End\ time_w + Ad\ Banner_w \qquad for\ website\ w\ (w = 1\ to\ 5)$$

GA Model: To Optimize Advertising Plan

The GA model is then fitted to find the best assignment of ad banners to websites and the start and end time of each ad banner display time. This GA model maximises the user clicks based on the predicted value from the regression model given the $300 budget constraint per day.

Results 1: Solving using Excel

Linear Regression Model

Table 2 Coefficients of Regression Model

Beta	*Coefficients*
Intercept	- 31,240.8251
W1 Start time	- 8,383.8886
W1 End time	8,903.3496
W1 Ad	900.1868
W2 Start time	- 12,460.0261
W2 End time	11,882.0000
W2 Ad	975.2972
W3 Start time	- 11,389.3928

Beta	*Coefficients*
W3 End time	10,470.6899
W3 Ad	2,313.9338
W4 Start time	- 10,214.0573
W4 End time	9,978.6532
W4 Ad	1,444.7968
W5 Start time	- 8,477.0341
W5 End time	9,208.9611
W5 Ad	1,659.0402

Table 3 Regression Statistics

Regression Statistics	
Multiple R	0.9552
R Square	0.9123
Adjusted R Square	0.9110
Standard Error	24,989.7017

The results of the regression model are given in Table 2 and Table 3. From the regression statistics, we see that the R square is 0.9123 which means the model explains over 90% of the variability of the data around its mean and that the model fits well.

GA Model

The GA model is built to maximize the predicted clicks by changing the ad banner number, start and end time (highlighted yellow in Table 4) for each website. The predicted clicks (calculation not shown in this report) are calculated from the coefficients and intercept of the regression model.

Table 4 Results from the Best Solution from Excel Solver

Website	Ad	Start Time	End time	Duration	Cost per hour	Cost
Website 1	4	0.52	0.54	0.02	$ 15.00	$ 0.29
Website 2	5	0.21	1.13	0.92	$ 10.00	$ 9.22
Website 3	3	0.32	23.99	23.67	$ 8.00	$ 189.38
Website 4	6	6.43	18.48	12.05	$ 8.00	$ 96.40
Website 5	2	13.48	13.86	0.39	$ 12.00	$ 4.64
Not assigned	1				**Total Cost**	**$ 299.93**

<u>Fitness Function</u>
Maximum of the predicted user clicks.

<u>Constraints</u>
The following constraints were put in place in the GA model:

- Total Cost ≤ $300
- Ad Number = integer
- Ad Number = All Different
- Ad Number ≥ 1
- End time ≥ Start time
- 0 ≤ Start time & End time ≤ 24

<u>Other Parameters</u>
Solving method = "Evolutionary"
Population size = 1,000
Mutation Rate = 0.075
Maximum time without improvement = 1,000

<u>Best Model</u>
We have run the solver multiple times, each time stopping only when it did not find a better solution after reaching the maximum time set. We noticed that the Solver gave significantly different results depending on the initial values that we set for the chromosome, hence the initial values are randomized for each run.

After multiple tries, our best solution is shown in Table 4 and the actual user clicks achieved is shown in Figure 3.

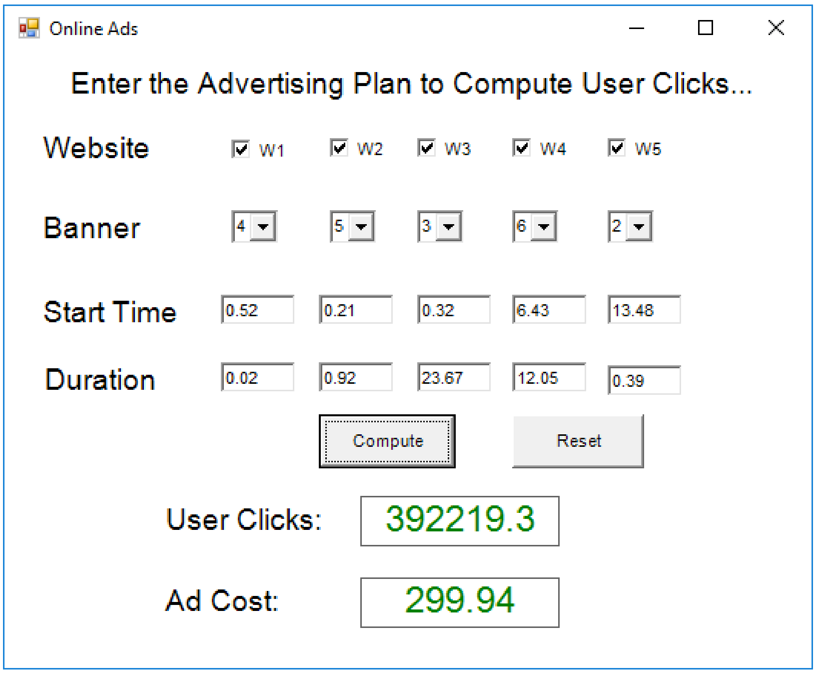


Figure 3 Workshop 1A – Actual User Clicks from Excel Solver Solution

From the actual results, we can see that the predicted and actual user clicks are quite close to each other, indicating the regression model indeed has a good fit to the data. The comparison between the actual and predicted user clicks is shown in Table 5.

Table 5 Actual vs Predicted User Clicks

	User Clicks
Predicted	387,170
Actual	392,219
Actual *minus* Predicted	5,049

Results 2: Solving using Java

As JGAP do not have options of specifying hard constraints to the optimization problem, all the constraints need to be specified as a part of fitness function only. JGAP only allows for range constraints to be specified beforehand. Same constraints as used in Excel were used. The fitness function for this implementation was designed to return 0 whenever any of the constraints were broken, thus essentially replicating a hard constraint. The other part of fitness function was similar to Excel.

Linear Regression Model

Similarly, all the 1,000 observations were used to fit a regression model. From the results shown in Table 6, we see that the linear regression model from Java are similar to the regression model fitted using the Excel Data Analysis tool, as what we would have expected. Weka library was used for the model.

Table 6 Regression Statistics

Regression Statistics	
Multiple R	0.9533
R Square	0.9088
RMSE	25,276.1695
MAE	20,194.6957

GA Model

Table 7 shows the best solution from Java and Figure 4 shows the actual user clicks achieved using this best solution.

Table 7 Workshop 1A Results from the Best Solution from Java

Website	Ad	Start Time	End time	Duration
Website 1	2	23.60	23.70	0.10
Website 2	3	0.30	0.50	0.20
Website 3	6	0.30	23.60	23.30
Website 4	4	0.30	13.90	13.60
Website 5	5	23.20	23.30	0.10

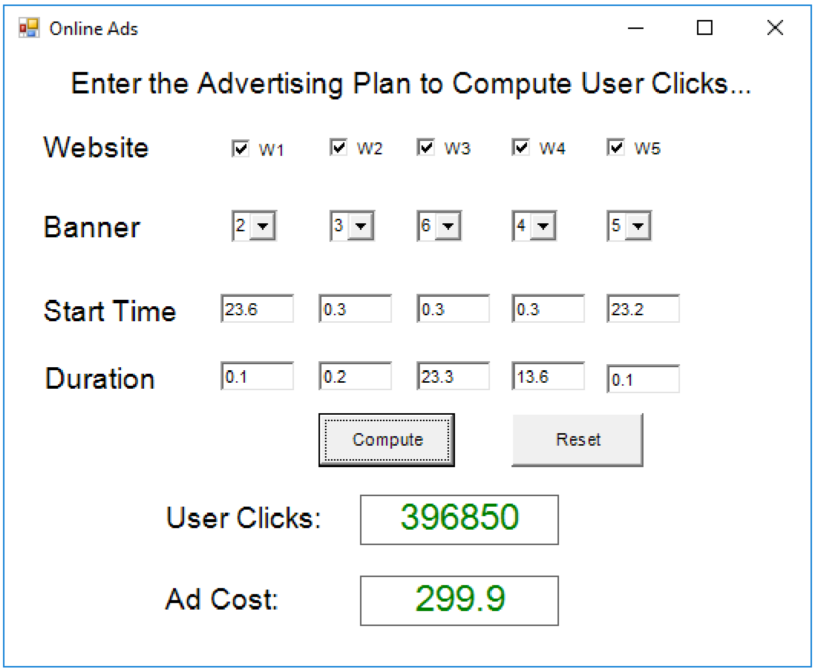


Figure 4 Workshop 1A – Actual User Clicks from Java solution

From the actual results, we can see that the predicted and actual user clicks are quite close to each other, given in Table 8. We can see that the regression model only slightly over-predicts the number of user clicks.

Table 8 Actual vs Predicted User Clicks

	User Clicks
Predicted	413,713
Actual	396,850
Actual *minus* Predicted	-16,863

Workshop 1A Results Comparison: Excel vs Java

Table 9 shows the comparison between the solution done using Excel and Java. The GA model from Java gave a higher number of actual clicks hence we conclude that the solution from Java is better among the 2.

Table 9 Comparison between Excel and Java

	Excel	Java
Regression: R square	0.9123	0.9088
GA Model: Predicted Clicks	387,170	413,713
Actual Clicks	392,219	396,850
Total Costs	$ 299.94	$ 299.90

WORKSHOP 1B

Background Knowledge

The number of user clicks depend on the time of day in which the ads are displayed, and a day is divided into 3 time periods within which the number of user clicks per hour is stable. The user clicks achievable is also modelled using the duration of the placement and the clicks per hour multiplied by a scale factor. With this information, we have redesigned the architecture of the system.

Architecture

The architecture of the hybrid intelligent system is shown in Figure 5.

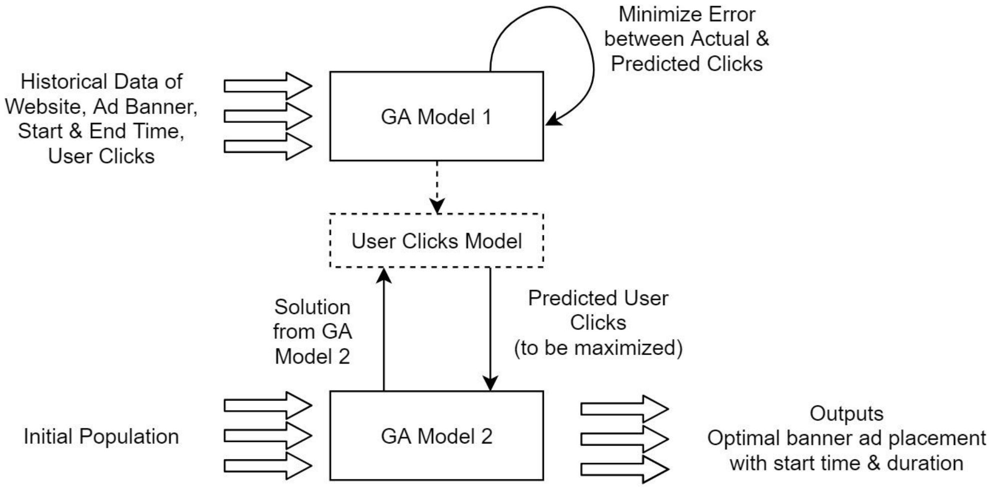


Figure 5 Workshop 1B Architecture

GA Model 1: To Predict User Clicks

Using the 1,000 observations, the first GA model is built to find the 2 cut-off points in a day, the clicks per hour for each duration and the scale factor for each banner ad. These parameters are used to predict the number of user clicks per day. The GA model minimises the difference between the actual and the predicted clicks in the given historical data.

GA Model 2: To Optimize Advertising Plan

The second GA model is then fitted to find the best assignment of ad banners to websites and the start and end time of each ad banner display time. This GA model maximises the user clicks based on the predicted value from the regression model given the $300 budget constraint per day. The user clicks is estimated from the parameters estimated from the first GA model.

Results 1: Solving using Excel

GA Model 1

All the 1,000 observations were used to build the first GA model to predict the number of user clicks by duration, the cut-off points and the scale factors for each banner.

Fitness Function
Minimum of the square of the difference between actual and projected user clicks.

Constraints
The following constraints were put in place in GA Model 1:

- Click for duration 1, 2, 3 ≤ 17,000 (iteration in '000)
- Click for duration 1, 2, 3 ≥ 0 (iteration in '000)
- CP2 ≥ CP1
- 0 ≤ CP1, CP2 ≤ 24
- Scale Factors ≤ 5

The clicks for duration 1 to 3 were different for each of the 6 websites. The clicks were iterated in thousands of clicks to reduce the search space. The clicks were also set to be less than 17,000. This is based on our initial data exploratory where the maximum clicks per hour are found to be 16,647 from the historical advertising data.

CP1s were also set to be larger than CP2s as one of the constraints to ensure that the second cut-points were at a later than the first cut-off points. Each scale factors were given a constraint of less than 5 and this is an arbitrary number as we do not have any knowledge about the range of the scale factor values.

Other Parameters
Solving method = "Evolutionary"
Population size = 2,000
Mutation Rate = 0.075
Maximum time without improvement = 200

Best Model

Table 10 Workshop 1B Excel – Best Results from GA Model 1

Website	CP1	CP2	Click1	Click2	Click3
Website1	1.00	5.00	6000	6000	1000
Website2	2.00	4.00	4000	13000	1000
Website3	0.00	7.00	5000	13000	1000
Website4	3.00	10.00	6000	8000	1000
Website5	0.00	11.00	4000	5000	1000

Ad Banner	1	2	3	4	5	6
Scale Factor	0.8689	0.7775	0.6870	0.4360	0.6956	0.9761

After many runs of GA Model 1 together with GA Model 2 with different initial values and population size, we decided that the model in Table 10 are the best GA Model 1 that we have. GA Model 2 is discussed in the following section.

GA Model 2

Fitness Function

Maximum of the predicted user clicks. The predicted user clicks are calculated from the cut-off points, clicks per hour for each website and the scale factors for each ad that was the output from the GA model 1, shown in Table 10.

Constraints

The following constraints were put in place in the GA model 2. These constraints are the same as the constraints set in Workshop 1 for the GA model.

- Total Cost ≤ $300
- Ad Number = integer
- Ad Number = All Different
- Ad Number ≥ 1
- End time ≥ Start time
- 0 ≤ Start time & End time ≤ 24

Best Model

The best model from GA model 2 is given below.

Table 11 Workshop 1B Excel – Best Results from GA Model 2

Website	Ad	Start Time	End time	Duration	Cost per hour	Cost
Website 1	2	9.28	12.79	3.52	$ 15.00	$ 52.73
Website 2	4	10.08	17.97	7.89	$ 10.00	$ 78.92
Website 3	6	0.07	10.65	10.58	$ 8.00	$ 84.62
Website 4	1	5.60	14.10	8.50	$ 8.00	$ 68.01
Website 5	3	8.71	9.98	1.27	$ 12.00	$ 15.28
Not assigned	5				**Total Cost**	**$ 299.56**

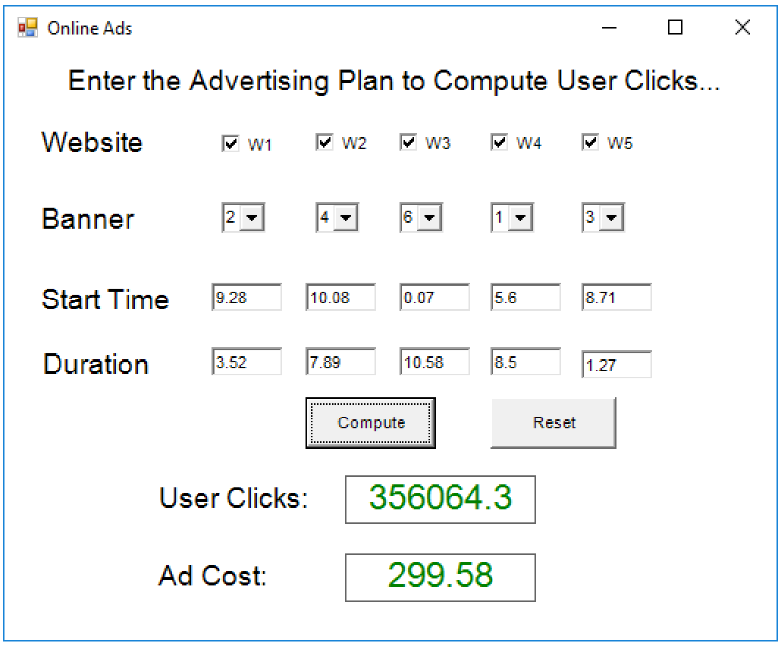


Figure 6 Workshop 1B – Actual User Clicks from Excel Solver solution

Table 12 below shows the actual and predicted user clicks. We can see that GA Model 1, in this case, did not give a good prediction of the user clicks as it severely under-predicted the number of user clicks. We have however chosen this version of GA Model 1 because the output from this model gave better results in GA Model 2, in terms of high actual user clicks.

Table 12 Actual vs Predicted User Clicks

	User Clicks
Predicted	136,197
Actual	356,064
Actual *minus* Predicted	219,867

Results 2: Solving using Java

We implemented the hybrid system using JGAP. Again as the hard constraints other than range cannot be specified in JGAP directly, they were transferred to the fitness function. The fitness function was modified to return 0, whenever any of the constraints failed. By returning the smallest possible value of fitness function, it can be made sure that the genomes with non-zero fitness values are not breaking any constraints.

GA Model 1

The major difference between this model and the GA Model 1 using Excel Solver is that the user clicks are iterated not by thousands, but at a more granular level, by the ones. We believe this leads to a better accuracy of the model. Also, the max evolution was set to 1000 generations and population size was set to 3000 genomes. The complete Java program takes nearly 30 minutes to complete. We feel that this large population time and longer evolution duration covers larger search space than the Excel solutions.

Table 13 Workshop 1B Java – Best Results from GA Model 1

Website	CP1	CP2	Click1	Click2	Click3
Website1	6.98	14.82	7,738	5,162	12,322
Website2	7.08	14.47	8,168	13,695	8,423
Website3	7.01	14.95	11,385	10,678	5,589
Website4	10.83	13.55	9,879	4,692	9,905
Website5	15.20	17.07	6,660	10,993	15,616

Ad Banner	1	2	3	4	5	6
Scale Factor	1.0030	0.9193	1.1044	1.0198	1.0306	1.2923

GA Model 2

Table 14 shows the best solution from Java and Figure 7 shows the actual user clicks achieved using this best solution.

Table 14 Workshop 1B Results from the Best Solution from Java

Website	Ad	Start Time	End time	Duration
Website 1	1	22.80	23.00	0.20
Website 2	4	9.90	16.00	6.10
Website 3	6	0.90	14.00	13.10
Website 4	5	14.80	23.20	8.40
Website 5	3	17.80	23.10	5.30

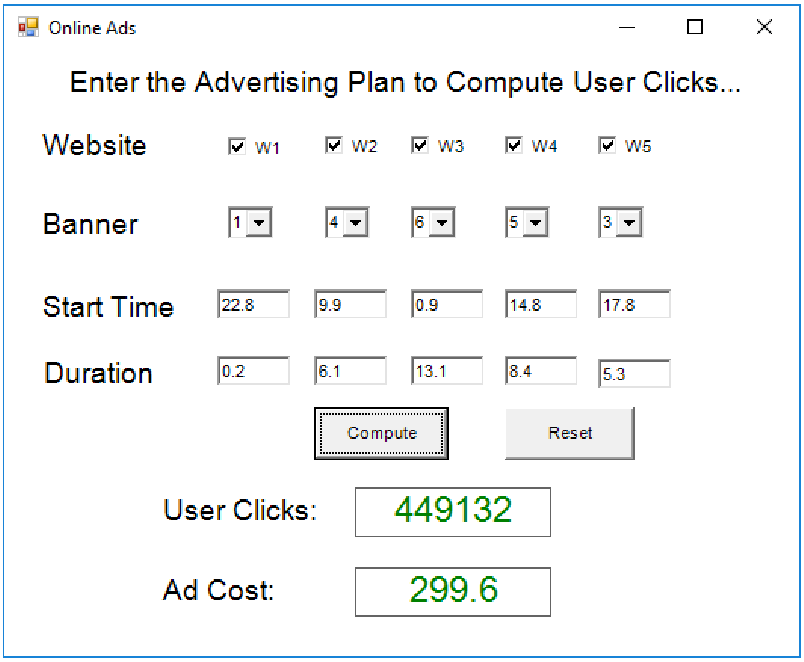


Figure 7 Workshop 1B – Actual User Clicks from Java solution

In comparison to GA Model 1 from Excel Solver, we see that the GA Model 1 produced by Java has better far better accuracy as it only slightly over-predicts the user clicks as compared to the significant under-prediction using Excel Solver.

Table 15 Actual vs Predicted User Clicks

	User Clicks
Predicted	502,880
Actual	449,132
Actual *minus* Predicted	-53,748

Workshop 1B Results Comparison: Excel vs Java

Table 16 shows the comparison between the solution done using Excel and Java. The GA model from Java gave a higher number of actual clicks hence we conclude that the solution from Java is better among the 2.

Table 16 Comparison between Excel and Java

	Excel	Java
Predicted Clicks	136,197	502,880
Actual Clicks	356,064	449,132
Total Costs	$ 299.56	$ 299.60

Comparison: Workshop 1A vs Workshop 1B

As a summary, we have compiled the actual user clicks from all the 4 models in Workshop 1A and 1B in Table 17. The Java solution from Workshop 1B gave the highest number of user clicks.

Table 17 Comparison of Actual User Clicks Count

	Excel	Java
Workshop 1A – Actual Clicks	392,219	396,850
Workshop 1B – Actual Clicks	356,064	449,132

From both workshops, we found that the additional background knowledge where number of clicks depends on the time of the day indeed improve the user clicks performance by about 52,000 clicks more per day, according to our Java model. Perhaps, we could improve the number of user clicks further by breaking down into more durations per day and each website having a different cut-off points.s

WORKSHOP 2B: FORECASTING

Direct mailing campaign for a bank

Objective

To build an intelligent hybrid system to generate a prospect list of 400 customers from a database of 4,000 customers that maximises the expected profit.

Tools

We have used python to solve this problem. Following are the main libraries used:

1. SkFuzzy: For the fuzzy inference system
2. PyEvolve: For genetic evolution
3. Keras: For neural network and training
4. Pandas, Scikit-learn, Numpy: For data wrangling and pre-processing.

Data Description

The data provided comprises:

1. a set of trial promotion results (Table 18) containing 1,000 observations
2. a set of customer data containing 4,000 observations.

In 1. (Table 18), each observation refers to a customer with the attributes *sex, marital status, age, number of children, occupation, education level, income level, average balance, average number of transactions* and *the decision* indicating whether the customer buys product A or B or None.

	Sex	mstatus	age	children	occupation	education	income	avbal	avtrans	decision
Index										
1	F	married	56.82	1	legal	secondary	3105.39	33003.48	1776.81	None
2	M	widowed	87.35	3	retired	tertiary	4874.08	18941.99	863.56	None
3	M	single	28.75	0	manuf	professional	14232.37	30013.32	3231.14	B
4	F	married	35.71	0	education	postgrad	3214.93	15423.24	1996.09	None
5	M	single	20.53	0	construct	tertiary	3214.93	15423.24	1996.09	None

Table 18 Sample of Trial Promotion Results

Architecture

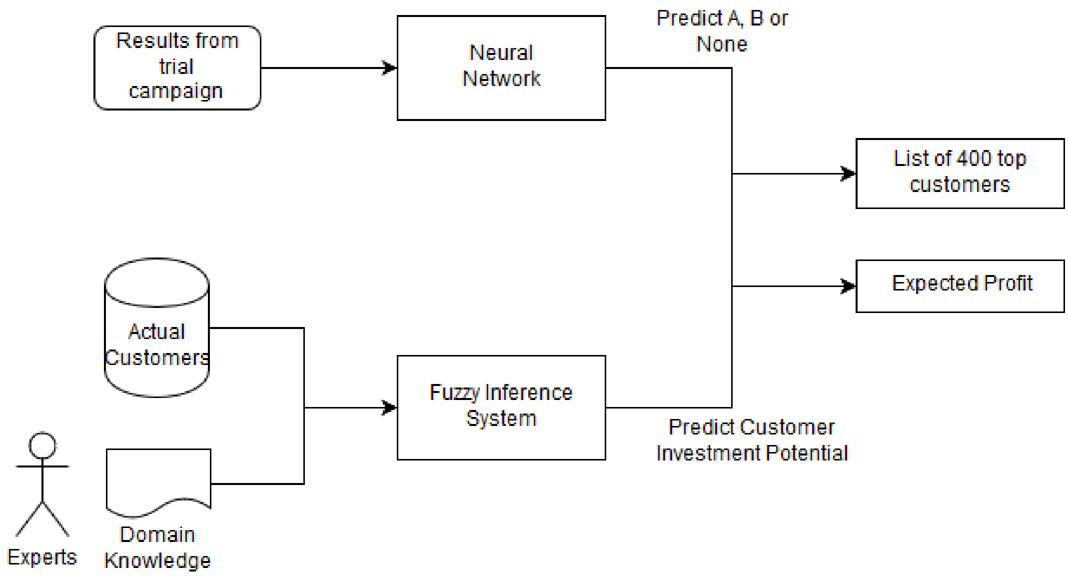


Figure 8 Independent sub-problem hybrid architecture

Neural Network

A neural network classifier serves as a customer purchase propensity model which predicts if a customer will purchase product A, B or none. The output variable for the classifier is 'decision'.

Dataset

The dataset (Table 18) is highly imbalanced relative to this output variable (Figure 9).

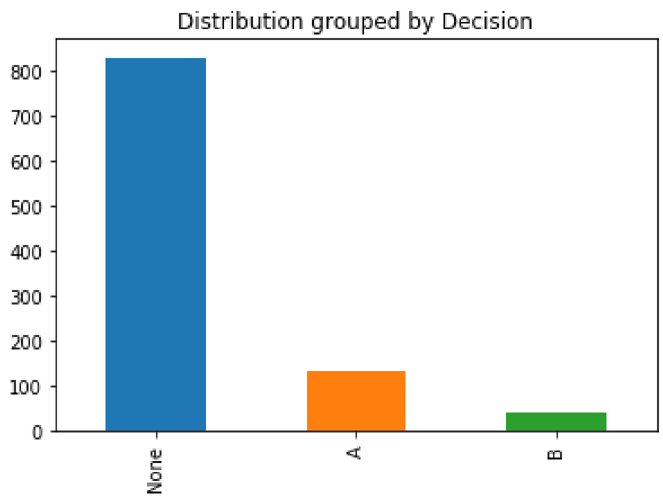


Figure 9 Distribution grouped by decision

Train / Test Split

A train and test dataset is created out of the 1000 observations using a stratified 80:20 split so that the distribution is preserved in both train and test dataset.

Data Transformation

In the train dataset of 799 observations, the ratio of observation in the 3 classes is A (14%), B(3.8%) and None (82.2%). To address this imbalance in the training data, the data is rebalanced according to the ratio 40:40:60. Due to the presence of categorical variables in the dataset, a naïve random oversampling method is used on class A and B. No under sampling is performed to avoid losing any data because the dataset is small.

After oversampling, the total observations are 1643 with 493 (A), 493 (B) and 657 (None).

The categorical input variables are one-hot encoded and the numerical input variables are scaled to mean 0 and standard deviation 1.

Base Model

A stratified 80:20 split is first performed on the train data to obtain a train and validation dataset.

A base model is trained to obtain an approximately good model relative to the metric. As the dataset is imbalanced, the metric chosen is the macro f1 score (average of the score of the 3 classes) as it is desirable to achieve a good precision-recall trade-off for each of the three classes A, B and None.

The base model is a multi-layer feedforward neural network of 2 hidden layers of 32 and 16 nodes respectively. The network converges quickly within 200 iterations especially for larger number of hidden nodes. To avoid overfitting, L2 regularisation with a regularization parameter of 0.001 was used in the loss function. Dropout layers with a parameter of 0.5 were added to the input layer as well as all the hidden layers. These measures help delay the onset of overfitting and allows the train and validation f1 scores to increase and converge.

The training converges within 2000 iterations (Figure 10).

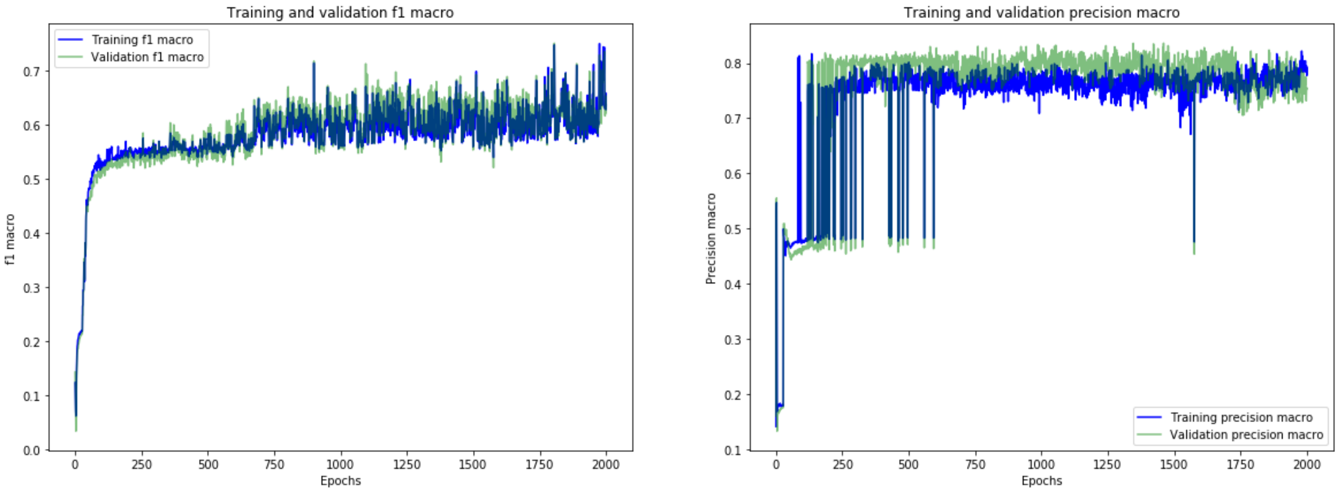


Figure 10 training and validation macro f1 and precision scores

The initial results show that mis-classification is biased to the majority class 'None' i.e. many 'None' observations were mis-classified as 'A' or 'B'. This is undesirable as it will have an adverse impact on estimated expected profits. A higher precision for 'A' and 'B' is preferred at the cost of recall. The model training is adjusted using the class-weights ratio of 'A':1, 'B':1, 'None':2.25 to adjust the training results.

The train and validation metrics and confusion matrices are as follows.

	class	precision	recall	f1-score	support
Train	A	0.86	0.13	0.23	330
	B	0.92	1.00	0.96	330
	None	0.59	0.92	0.72	440
	macro avg	0.79	0.68	0.63	1100
Validation	A	0.77	0.15	0.25	163
	B	0.91	1.00	0.95	163
	None	0.58	0.89	0.70	217
	macro avg	0.75	0.68	0.63	543

Table 19 Base Model Training and Validation Metrics

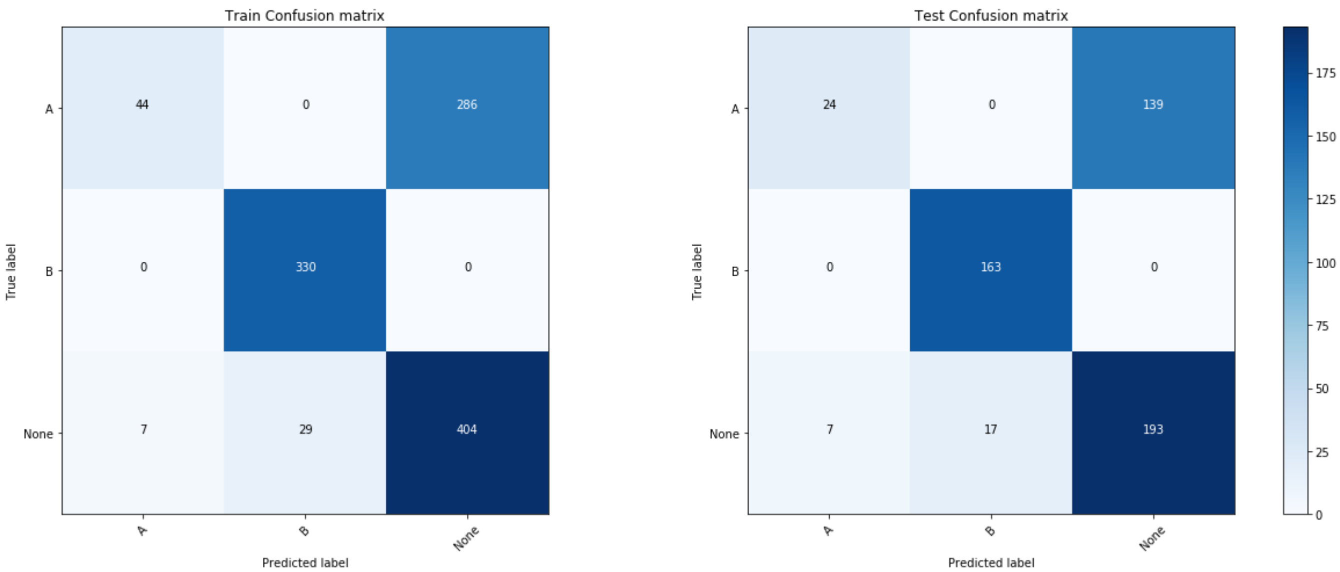


Table 20 Base Model Training and Validation Confusion Matrix

Search for Better Model using Cross-Validation

The entire train dataset was used for training using stratified cross-validation.

A search for the best model using the set of model parameters was performed:

1. layer1: 16, layer2: 16, layer3: 0
2. layer1: 32, layer2: 16, layer3: 0
3. layer1: 32, layer2: 32, layer3: 0
4. layer1: 32, layer2: 16, layer3: 8
5. layer1: 32, layer2: 32, layer3: 8

The best model using the f1 score as the metric, is between the model 2 and 3 with slightly different results over multiple runs. The model 3 (32, 32) is selected as the best model.

Model	Cross-validation macro f1 score
layer1: 16, layer2: 16, layer3: 0	0.65
layer1: 32, layer2: 16, layer3: 0	0.82
layer1: 32, layer2: 32, layer3: 0	0.83
layer1: 32, layer2: 16, layer3: 8	0.57
layer1: 32, layer2: 32, layer3: 8	0.63

This best model is retrained on the entire train dataset and scored in the test dataset. The metrics and confusion matrices are as follows:

	class	precision	recall	f1-score	support
Train	A	0.79	0.94	0.86	493
	B	0.92	1.00	0.96	493
	None	0.94	0.75	0.83	657
	macro avg	0.88	0.89	0.88	1643
Test	A	0.31	0.67	0.42	21
	B	0.11	0.33	0.17	9
	None	0.90	0.68	0.77	170
	macro avg	0.44	0.56	0.45	200

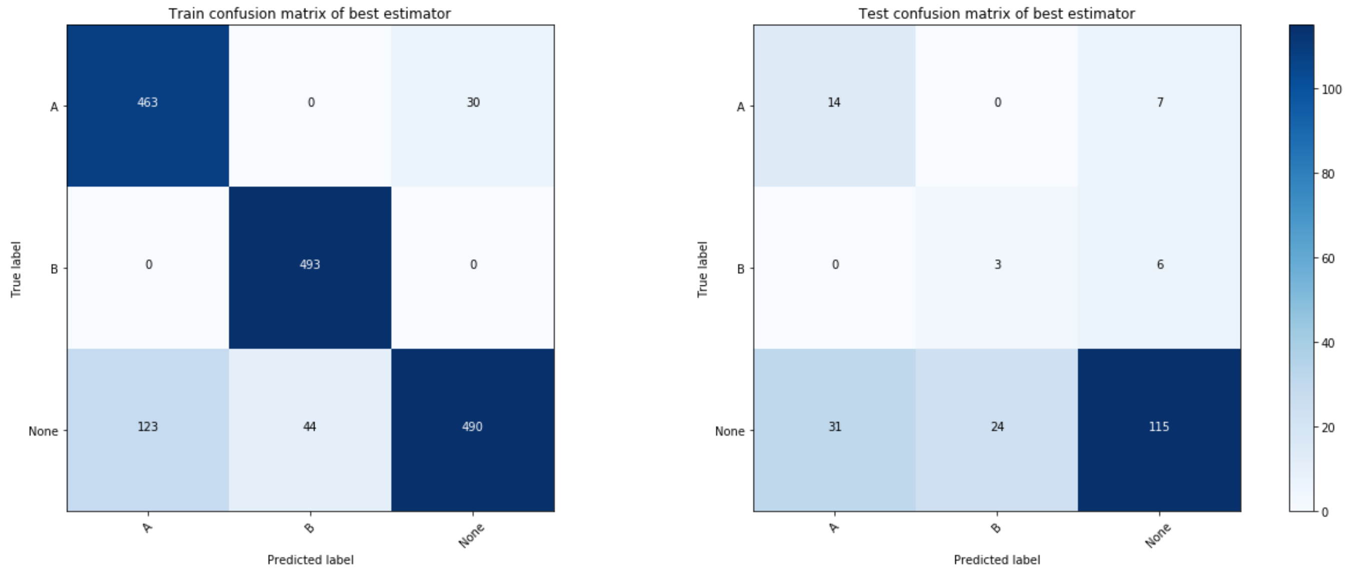


Figure 11 Best Model Train and Test Confusion Matrix

The test results are much lower than the train results. This could be because the test dataset is small. A better estimation of generalisation error could be obtained if more data was available.

Precision and Recall Trade-off for Decision Threshold

The precision and recall curves for class 'A' and 'B' show that setting the decision threshold at 0.5 is optimal for precision vs recall for the class 'A' and 'B'.

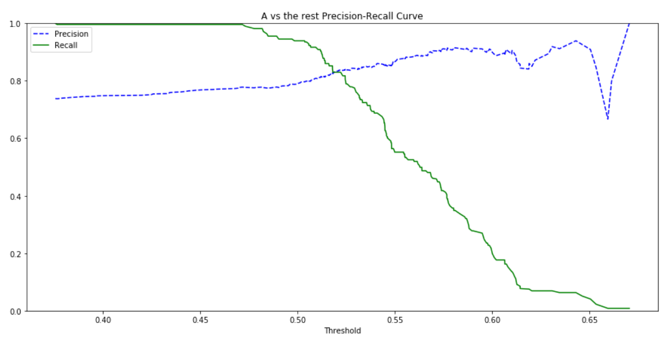


Figure 12 Precision and Recall Curve Class 'A'

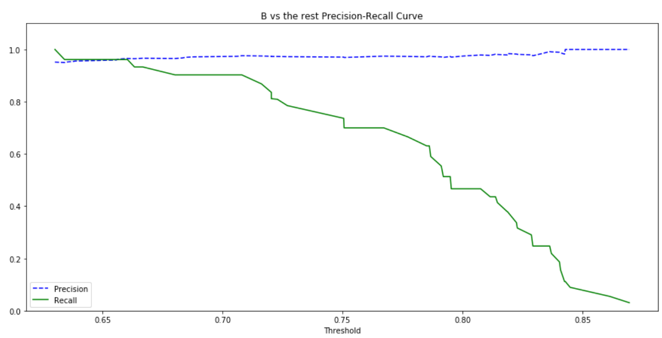


Figure 13 Precision and Recall Curve Class 'B'

Final Model

The best model is retrained using the entire dataset of 1000 observations.

Scoring the Final Model on the Customer Database

The final model is used to predict the 'decision' for the customer database of 4000 observations. The prediction probability is generated as well. A check of the prediction probability showed that only 3 predictions were under 0.5. Hence, the predictions were not modified by adjusting using the decision threshold.

Fuzzy Inference System

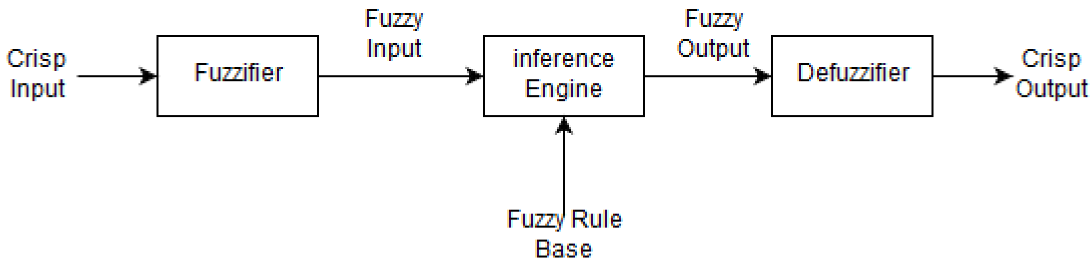


Figure 14 Fuzzy Inference System

Skfuzzy provides an implementation of the Mamdani inference system. The implementation mainly comprises three methods:

1. **Aggregation**: This finds the net accomplishment of the antecedent by AND-ing or OR-ing together all the membership values of the terms that make up the accomplishment condition.
2. **Activation**: The degree of membership of the consequence is determined by the degree of accomplishment of the antecedent.
3. **Accumulation**: Apply the activation to each consequent, accumulating multiple rule firings into a single membership value. Centroid method is used to combine all the activation and return a single de-fuzzified value.

Membership Functions
Figure 16 to Figure 23 below shows the membership function that were used in our initial model.

- Variables: age, income, avtrans, avbal
  For the continuous variables, we have used the statistics from the training data to guide the estimation of the parameters of the membership functions. To simplify the problem, these variables are assumed to have 3 levels: low, medium and high.

  From the training data, we first computed the first quartile, median and third quartile figures for each of the continuous variables. We then derived the membership function based these 3 figures shown in Figure 15.

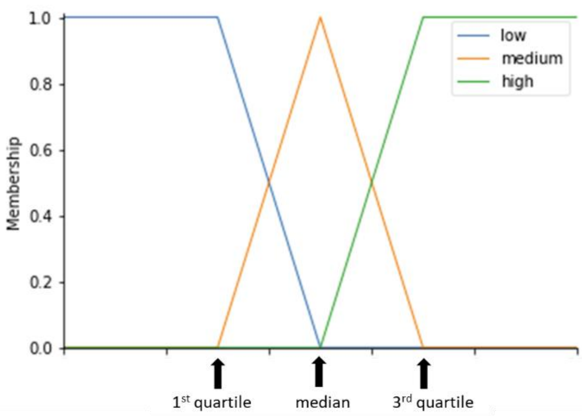


Figure 15 Membership Function Design for Continuous Variables

- Variables: children, education, customer investment potential (CIP)
  The parameters for these variables were more of a guess-estimate based on our own knowledge and assumptions as there were no data to guide the setting of the levels for these parameters. Some of our rationales are:
  - Number of children ≤ 1 have low membership function
  - Number of children = 4 have high membership function
  - For education level, we assigned 0 for secondary, 1 for tertiary, 2 for professional and 3 for postgrad. 0 and 1 have low membership function while 2 and 3 have high membership function
  - Similarly, for CIP, for the range of score from 0 to 10, naturally 5 becomes the median and above 5 is considered high while below 5 is conserved low

- Variables: occupation,
  As this is a categorical variable, we have assigned value 1 to 8 for the 8 occupations in no particular order.

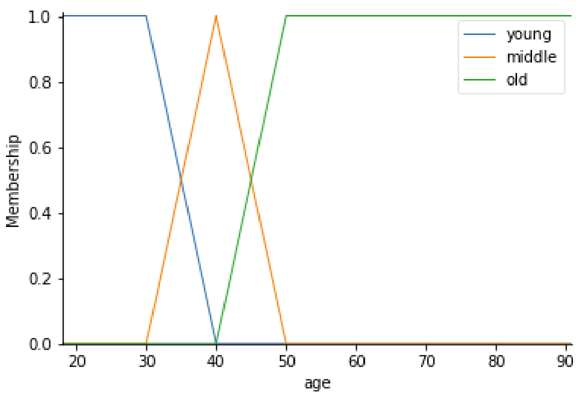


Figure 16 Membership Function: Age

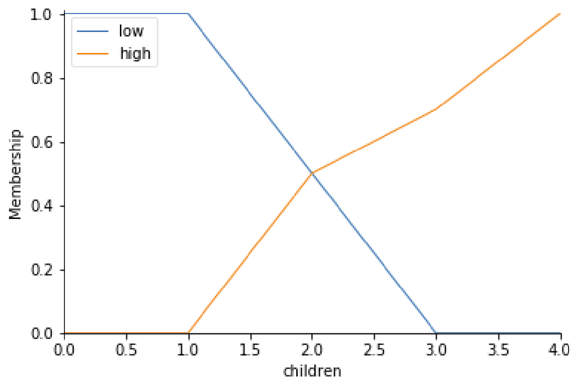


Figure 17 Membership Function: Children

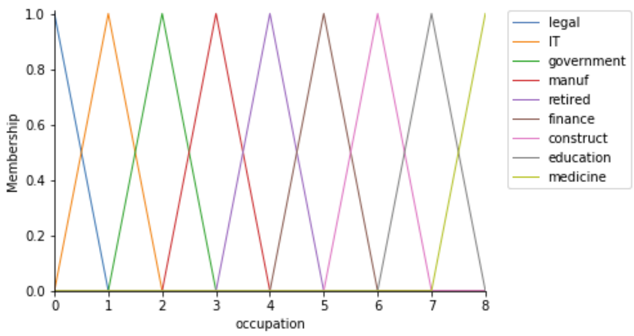


Figure 18 Membership Function: Occupation

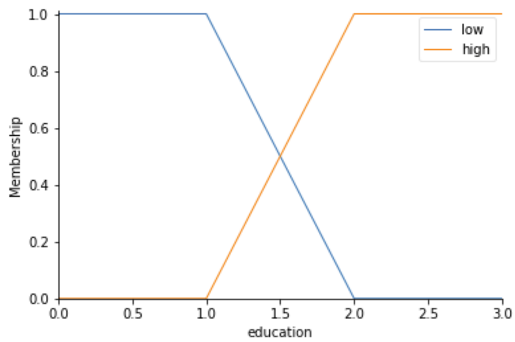


Figure 19 Membership Function: Education

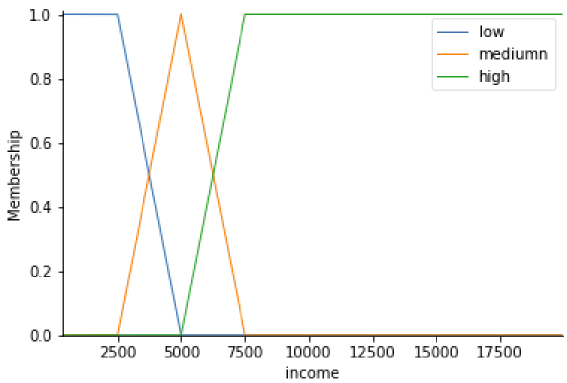


Figure 20 Membership Function: Income

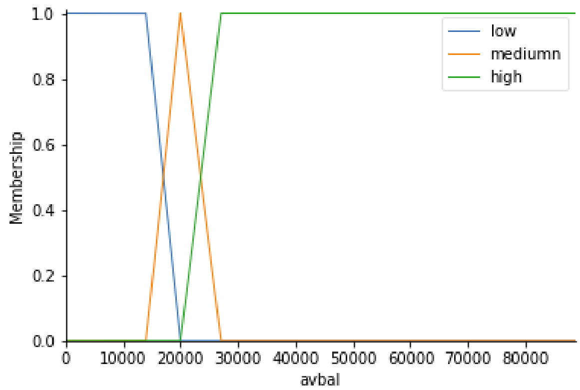


Figure 21 Membership Function: Average Balance ("avbal")

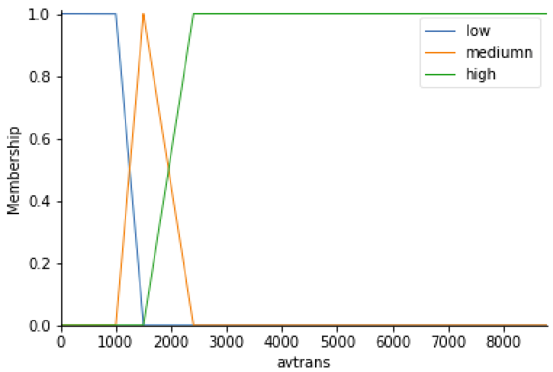


Figure 22 Membership Function: Average Transactions ("avtrans")

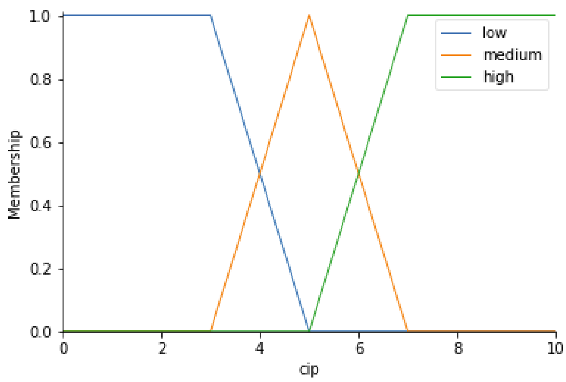


Figure 23 Membership Function: Customer Investment Potential (CIP)

Following fuzzy rules were derived from domain knowledge:

1. If available balance is high and available transaction is high then CIP is high
2. If available balance is medium and available transaction is high then CIP is medium
3. If available balance is high and available transaction is medium then CIP is medium
4. If available balance is medium and available transaction is medium then CIP is medium
5. If available balance is low and available transaction is low then CIP is low
6. If sex is male then CIP is high
7. If sex is female and marital status is single then CIP is high

8. If income is high then CIP is high
9. If age is middle then CIP is high
10. If occupation is retired then CIP is high
11. If occupation is either legal or medicine or education or finance or IT then CIP is high
12. If education is high then CIP is high
13. If education is high and age is middle then CIP is high
14. If income is high and age is old then CIP is high

These rules will form the rule base system for the Mamdani inference system.

Results

Classification Report

The classification results on 4000 customers are obtained by predicting using the trained NN model.

	class	precision	recall	f1-score	support
Test	A	0.32	0.32	0.32	498
	B	0.21	0.51	0.29	199
	None	0.86	0.78	0.81	3303
	macro avg	0.76	0.71	0.73	4000

The confusion matrix for the prediction is as follows:

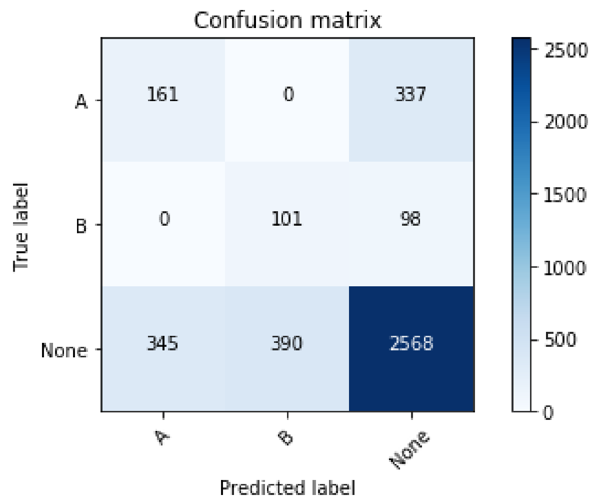


Figure 24 Confusion Matrix for Test Set (4000 customers) using NN model

CIP Prediction

The inference is used to predict CIP values for the 4000 customers from actual database.

Mean Absolute Error	3.140
Mean Square Error	10.893
Root Mean Square Error	3.300

Expected Profit

Expected profit for campaign is calculated by summing the expected profit of individual customer which will be mailed for the campaign.

Calculating the Actual Expected Profit
The file Cust_Actual.csv contains ground truth prediction for all the 4000 customers. Of which 498 customers have purchased A, 199 customers have purchased B and remaining customer did not make any purchase. Corresponding CIP value are also provided for each customer. To calculate the expected profit of each customer following formula is used:

$$\text{Expected profit* for customer}_i \begin{cases} = \text{customer investment score} * 0.6 & \text{if product purchased} = \text{A} \\ = \text{customer investment score} & \text{if product purchased} = \text{B} \\ = 0 & \text{if no product purchased} \end{cases}$$

Now each of customers are sorted in descending order of expected profit value. Top 400 customers are selected from this sorted list and expected profit of campaign is calculated using:

$$\text{Expected profit for campaign} = \sum_{\text{customers}} \text{Expected profit for customer}_i$$

Thus, value of actual expected profit is 1237.94.

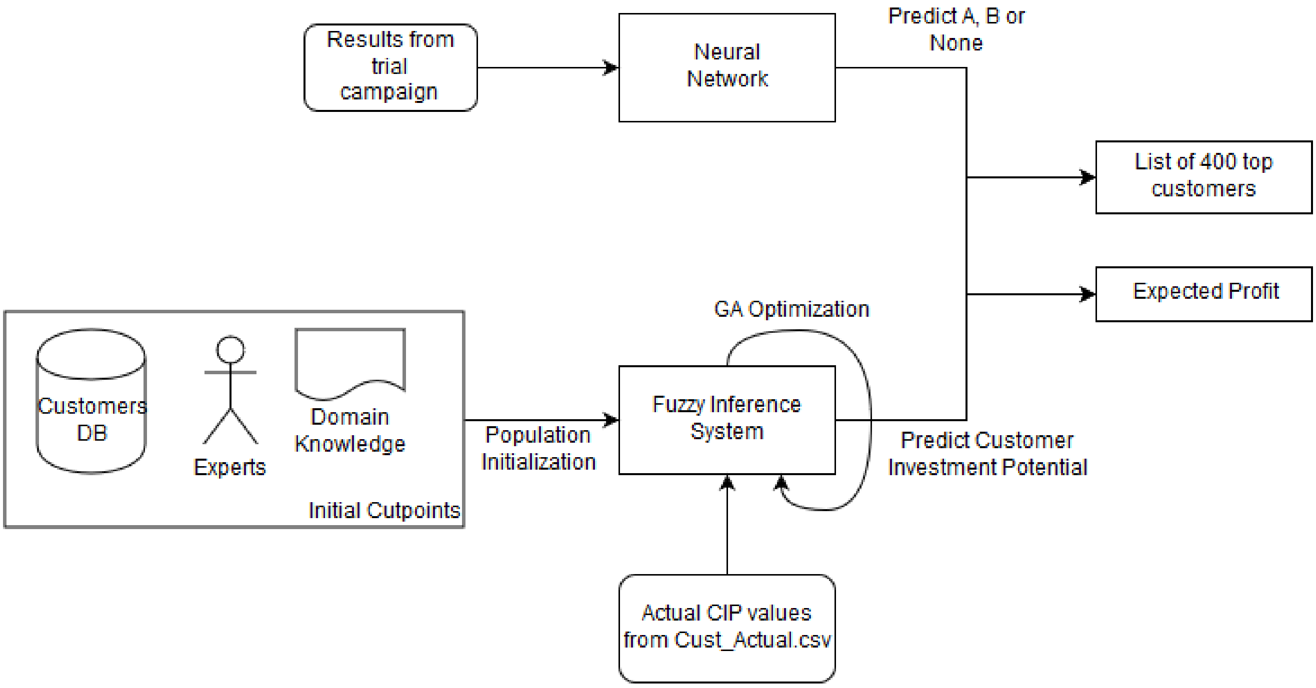
The prediction from inference system and neural network is used to calculate the expected profit from predicting 400 top customers. The top 400 customers are selected by following the same strategy as explained above.

Actual Expected Profit	1237.94
Predicted Expected Profit	2970.91
Difference (Actual – Predicted)	-1732.97

Further Improvements

Tuning the membership function cut-points using GA

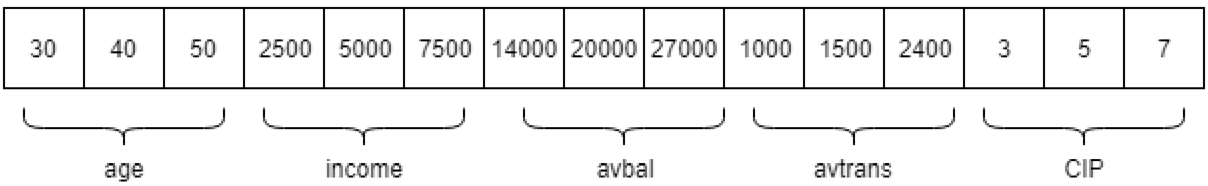
We used the genetic algorithm to find better cut-points for membership function. The population was initialized with chromosome built from the previous cut points (which were derived using statistical methods like mean, median & quartile).



25 Improved architecture with self-tuning fuzzy system

The various components of GA are described below:

- Chromosome Structure: The chromosome is a one-dimensional list which is built from cut points of membership functions in the fuzzy inference system. The membership functions included are age, income, avbal, avtrans and CIP; each having three cut points. Thus, the chromosome consists of 15 genes. The initial chromosome is set using the cut points used in base solution.



26 GA Chromosome Initialization

- Constraints: Range constraints are set for each gene of chromosome. For ranges, a set of alleles is created with each allele range set to minimum and maximum value of the corresponding membership function. This minimum and maximum values for each membership function are obtained from the dataset.

- Fitness function: The fitness function is defined as reciprocal of the sum of absolute error between actual CIP value and predicted CIP value using the cut point generated by the GA.

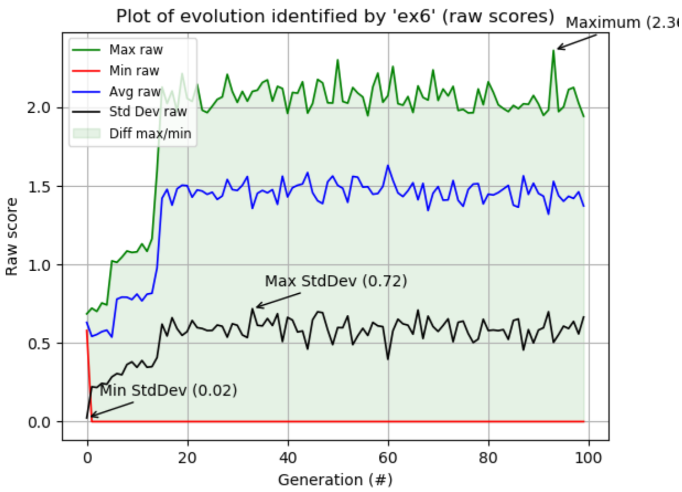
$$fitness\ function = \frac{100}{\sum_{i=1}^{50} |actual\ CIP_i - predicted\ CIP_i|}$$

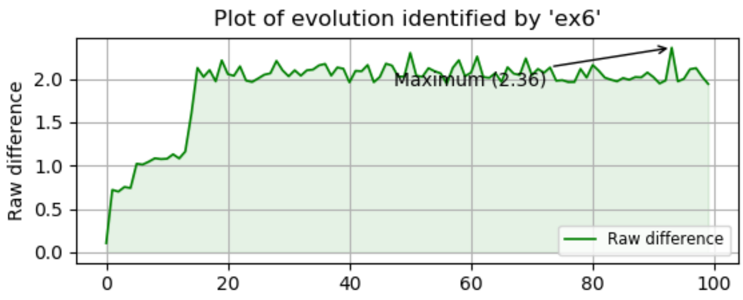
The error is calculated by summation of 50 random samples from 4000 records. This was done to improve the efficiency of GA. The execution time for fuzzy inference for single chromosome was around 12 – 13 seconds on 4000 records. Hence, if we use all 4000 records then each generation of 100 chromosomes will take a long time. Therefore, for each chromosome a random sample of 50 records were used, which took an execution time of 0.6 seconds per chromosome. A hard constraint assigns zero fitness score to above fitness function for the case when the cut points among the same membership function are not in increasing order.

- Other Initialization Parameters

Population Size	100
Number of generations	100
Mutation Rate	0.02
Crossover Rate	0.90
Elitism	True
Elitism Replacement	1
Selector Method	Rank Selector

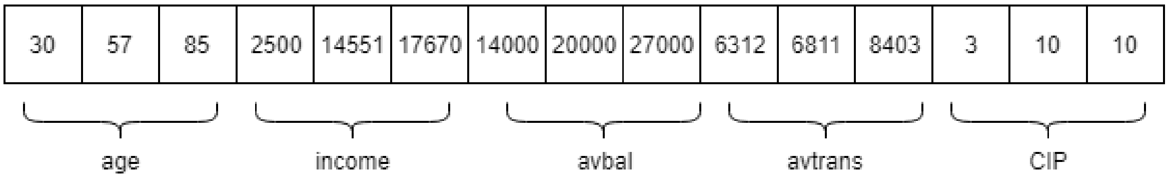
- GA Search process fitness plots





The plot for fitness function shows that there is sharp change between in average fitness at around 16 generation. After which average fitness value oscillates around 1.50. Also, the reason behind such a large difference between minimum fitness value and maximum value is because in each generation there will be chromosomes which will break the constraint, thus get 0 fitness raw score.

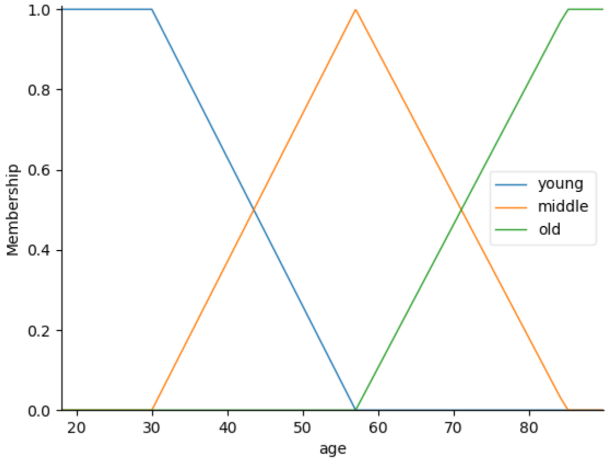
- Final Chromosome Structure: The GA is executed for 100 generations and the final chromosome structure is used as cut points for the fuzzy inference system.

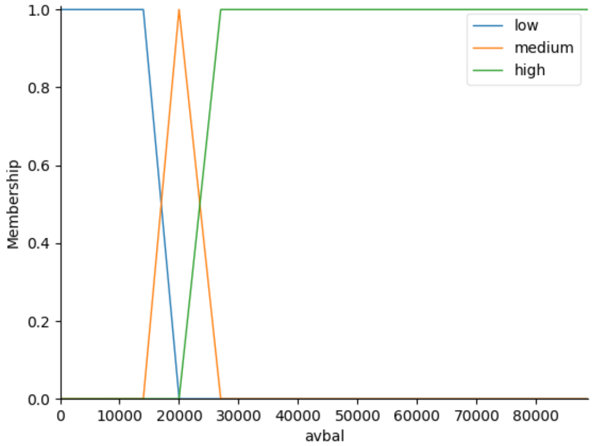


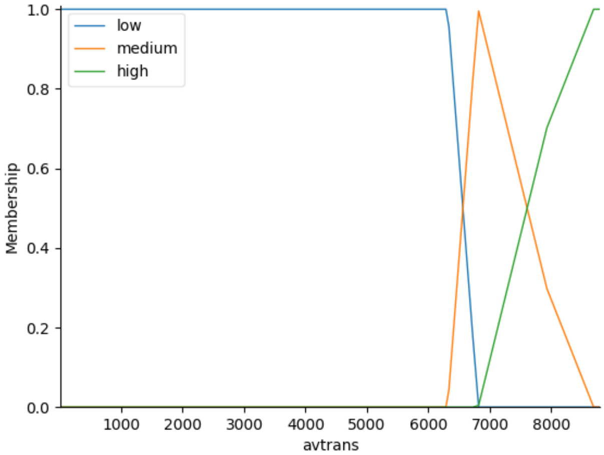
27 Final GA Chromosome Structure

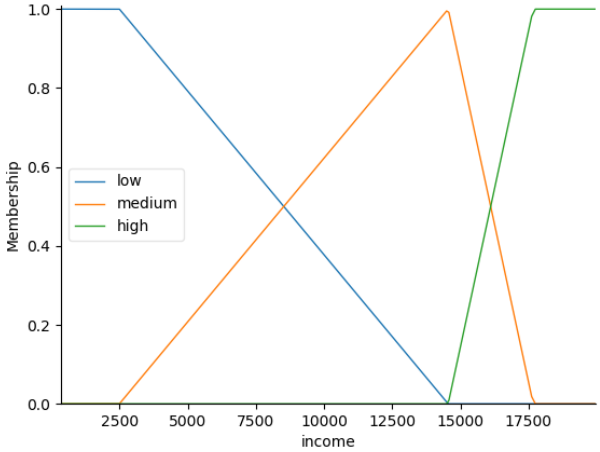
Optimized Membership Function

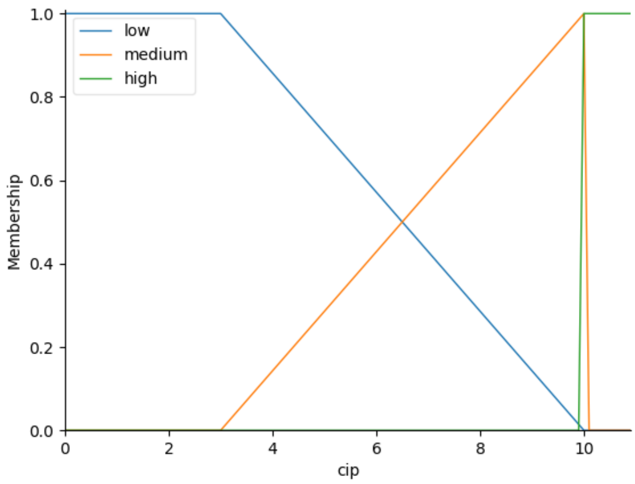
Following are the new membership function cut-points found using GA optimization.











For CIP the new membership function has been reduced to only two levels, as low and medium + high collapsed to form one label.

Improved Results

CIP Prediction

The CIP value predicted by the fuzzy inference system using the cut points obtained by GA has low error value. The Mean Absolute Error reduced from 3.140 (using base solution) to 1.166. Similarly, Root Mean Square Error improved from 3.300 (using base solution) to 1.337.

Mean Absolute Error	1.166
Mean Square Error	1.787
Root Mean Square Error	1.337

Expected Profit

The expected profit obtained using the improved architecture has low error value. The difference between the actual and predicted expected profit has reduced from 1732.97 (using base solution) to 546.13.

Actual Expected Profit	1237.94
Predicted Expected Profit	1784.07
Difference (Actual – Predicted)	-546.13

Improving Prediction Model by Adding Feature

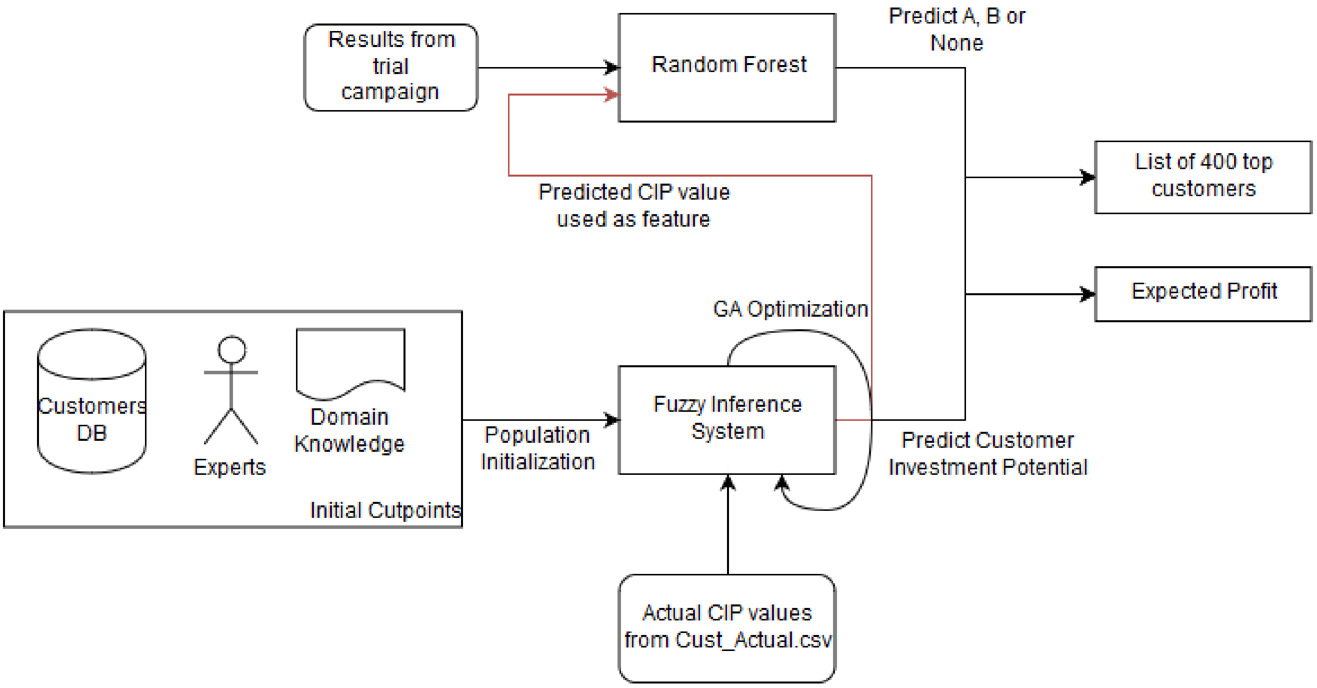
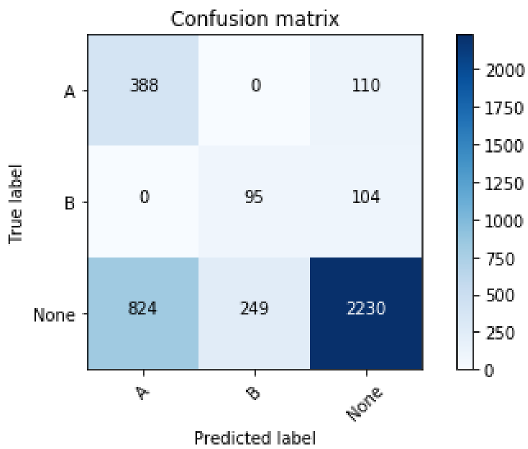


Figure 28 Cooperating Expert hybrid architecture

A random forest classifier with 500 trees was trained. The classification results on 4000 customers are obtained by predicting using the trained Random Forest model.

	class	precision	recall	f1-score	support
Test	A	0.32	0.78	0.45	498
	B	0.28	0.48	0.35	199
	None	0.91	0.68	0.78	3303
	macro avg	0.81	0.68	0.71	4000

The confusion matrix for the prediction is as follows:



29Confusion Matrix for Test Set (4000 customers) using Random Forest model

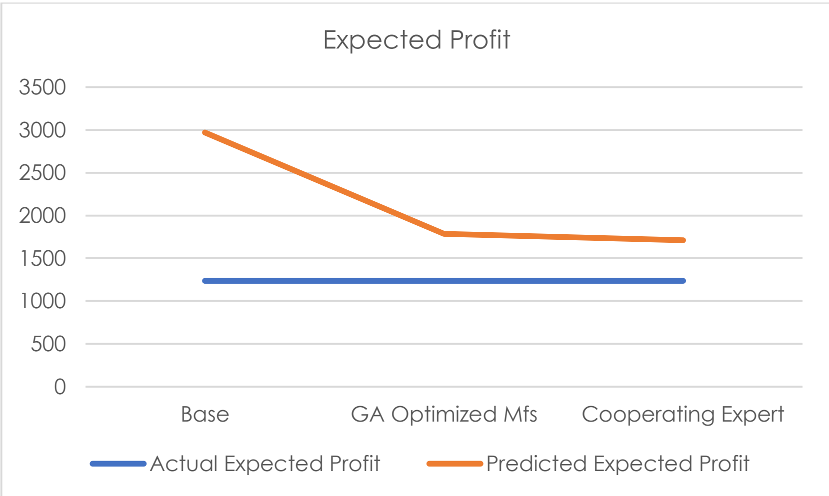
The CIP values generated from the fuzzy inference system are propagated to random forest classifier as a new feature. This feature combined with the features from original dataset are used to train the Random Forest Model. The expected profit obtained using the improved architecture by adding additional predicted CIP feature has low error value. The difference between the actual and predicted expected profit has reduced from 1732.97 (using base solution) to 474.42.

Actual Expected Profit	1237.94
Predicted Expected Profit	1712.36
Difference (Actual – Predicted)	-474.42

Summary

We tried three the following three hybrid architectures:

1. Base: Fuzzy Inference System + Neural Network
2. GA optimized Membership Functions: Inference System with optimized membership function + Neural Network
3. Cooperating Expert: Inference System with optimized membership function + Random forest classifier with CIP as an input from fuzzy system



Out of these later hybrid with optimized membership function performed significantly better than the first one. The cooperating expert performed marginally better with even lower absolute error. Shape of a few membership function was modified considerably when optimized with GA, suggesting that the initial cut points based on statistics and common knowledge were not good enough. Finally we were able to identified top 400 customers from cooperating-experts hybrid system which should be contacted by email during campaign to have higher expected profit. The final list is attached in Appendix A.

Appendix A

Indexes of 400 customers which needs to be contacted for email campaign. A CSV (top_400_customers.csv) file with index, decision, CIP and expected profit of the top 400 customer is attached with the submission.

1024	1514	1963	2531	3216	3609	4148	4652
1056	1515	1970	2537	3231	3615	4160	4656
1065	1528	1972	2557	3243	3627	4163	4659
1075	1543	1978	2558	3256	3631	4168	4660
1090	1544	1988	2560	3260	3633	4169	4667
1092	1546	2020	2567	3272	3638	4177	4668
1101	1558	2033	2569	3291	3654	4185	4679
1103	1560	2081	2572	3297	3668	4194	4681
1107	1561	2129	2574	3298	3676	4197	4682
1125	1564	2131	2579	3303	3707	4232	4685
1130	1567	2144	2581	3308	3711	4236	4687
1134	1568	2148	2612	3309	3720	4237	4693
1143	1577	2166	2624	3311	3729	4247	4695
1159	1582	2173	2648	3312	3738	4249	4696
1171	1599	2176	2684	3316	3743	4263	4701
1197	1602	2178	2704	3317	3745	4267	4704
1201	1603	2179	2713	3318	3746	4283	4710
1214	1611	2181	2749	3325	3753	4284	4715
1215	1617	2185	2795	3327	3754	4291	4718
1218	1620	2199	2802	3330	3764	4295	4719
1247	1622	2202	2815	3331	3772	4330	4720
1292	1624	2206	2817	3339	3774	4337	4723
1315	1636	2208	2819	3346	3778	4351	4724
1318	1639	2245	2827	3347	3782	4362	4725
1328	1654	2254	2838	3354	3809	4363	4729
1341	1671	2263	2844	3359	3815	4368	4730
1378	1688	2267	2847	3366	3820	4380	4734
1384	1701	2269	2877	3379	3831	4386	4736
1409	1726	2270	2883	3388	3847	4387	4749
1437	1750	2272	2897	3396	3879	4444	4751
1439	1755	2276	2902	3401	3885	4454	4776
1440	1756	2285	2918	3408	3892	4456	4778
1441	1757	2290	2928	3420	3899	4459	4798
1442	1759	2307	2950	3441	3905	4487	4800
1450	1761	2356	2951	3453	3910	4494	4803
1455	1773	2357	2969	3455	3930	4498	4821
1480	1778	2364	2970	3459	3955	4499	4834
1485	1784	2376	3009	3460	3958	4514	4846
1486	1788	2389	3020	3471	3965	4518	4860
1488	1790	2394	3034	3499	4003	4559	4871
1490	1806	2403	3042	3522	4005	4563	4879
1491	1815	2406	3057	3524	4022	4564	4883
1492	1823	2443	3063	3555	4034	4572	4890
1496	1839	2465	3065	3582	4041	4595	4899
1497	1858	2472	3089	3586	4091	4605	4911
1499	1859	2484	3132	3588	4093	4610	4935
1504	1871	2491	3147	3592	4094	4611	4941
1505	1916	2506	3164	3594	4111	4632	4948
1508	1926	2512	3169	3601	4118	4642	4969
1510	1956	2515	3205	3604	4135	4650	4987