A Genetic Programming-Based Approach for Building Automated Deception Detection Predictive Model

MIS40980 Natural Computing & Applications 2015

Mina Victor / 14200381 / mina.victor@ucdconnect.ie

1 Abstract

Genetic Programming (GP) emerges from being a standalone tool to solve and model optimisation problems, to be part of a solution in multi-disciple application. Combining GP with text mining for discovering unseen patterns in unstructured data aim to put earlier statistical approaches in question. In this paper, we experiment GP in building Automated Deception Detection (ADD) predictive model, one of recent application of text mining based on calculate set of linguistic cues. Our final goal is have (set) of hypothesis that distinguish truthful from deceptive statement.

Keywords: Genetic Programming, deception, deception detection, linguistics based cue, natural language processing

2 Introduction

Automated Deception Detection (ADD) [3] allows creating computational tool that is capable of alerting potentially deception content in an unstructured text given by a user, this is done without the need of manual intervention that in some situation impractical with high volume of text. Finding new approaches for detection is a challenging task that depends on, the following:

- **Context**: benefit clams, court witness, product & services reviews … etc.
- **Domain**: finance, insurance, social welfare … etc.
- **Median**: Scriber, online include emails & websites, pen & paper … etc.

With different combination of the context, domain and median, human linguistic behaviour varies. By using an adaptive natural computing techniques, like Genetic Programming (GP), we can provide the flexibility in the process of developing an automated deception detection model.

In this paper, we describe result of applying Genetic Programming as a model induction tool in the application of Automated Deception Detection, and using GP in conjunction with text mining [2] methods for linguistic cues feature extractions that is computed from dataset textual observation. In this experiment, we are trying to answer question the *“efficiency of Genetic Programming in predictive model induction for Automated Deception Detection? “*

In Section 2 we are going to present a literature review on different automated deception detection methods with the underlying hypotheses. In section 3, we will discuss the method used involving the workflow, GP operations, sample of tree model and stopping criteria. Then in Section 4, we will provide the details of the experiment with the parameter used and each run result. Finally we conclude in Section 5 with a discussion of our result.

3 Literature review

In this section, we discuss deception detection from two point of views, from psychological and computational.

3.1 Deception in Psycholinguistic

Deception detection has been early studied in psycholinguistic literature [8], in which it focuses on the psychological and neurobiological factors that enable humans to acquire, use and understand language. Studies shows that when people are attempting to construct a false story, they tend to:

- Use less complex language, due to consumption of cognitive resources by keeping the coherency of the story.
- Use more word reflecting negative emotion, due to feeling of guilt.
- Avoid statements of ownership as story tellers wants to distance themselves.

Keeping that in mind, many methods have been developed relaying on the mentioned hypotheses.

3.2 Automated Deception Detection

Content-Based Criteria Analysis (CBCA) is a technique developed to verify the veracity of a child's testimony in sex-crime cases [3]. It is based on the hypotheses that a statement based on fantasy will differ in quality and content from a statement based on actual experience. Trained evaluators judge the presence or absence of 19 criteria. The presence of each criterion suggests that the statement was derived from an actual experience, and is therefore not deceptive. CBCA hypothesizes that truthful messages will contain more unusual details, more superfluous details, more details overall, and more references to time, space, and feelings than deceptive messages as actual memories of an experience should contain more contextual details than deceptive statements

Management Obfuscation Hypothesis (MOH) is used in fraud detection of financial reporting [3]. As when companies perform poorly, management has an incentive to cover up this poor performance to delay stock price reaction by decreasing the readability of their annual reports. MOH hypothesis that, information that is more costly to extract and process will not be reflected immediately in market prices. Therefore, management can delay stock price reaction by obfuscating financial reports making them more costly to analyse. MOH uses SMOG reading index to operationalize and measure readability. The SMOG reading index takes into account sentence complexity (as measured by average sentence length) and word complexity (as measured by average word length) to determine readability. Longer sentences and longer words are a surrogate measure for complexity and should occur when fraud is present.

Reality Monitoring (RM) is a method that attempts to distinguish between memories based on true experiences from internally generated falsehoods or imagination [3]. RM hypothesizes that statements based on true memories and statements based on falsehoods differ in, the amount of perceptual details, the amount of contextual information and the quantity of cognitive operations described in the statements.

Others methods includes: Scientific Content Analysis (SCAN), Interpersonal Deception Theory and Four Factor Theory (FFT) [3], where each approach define a set of language cues and then drive a statistical model induction based on stated hypothesis.

This paper focus on deriving deception detection model that is based on the linguistic cues, but using a natural computing method as the underlying method and not statistic approach.

4 Method

Deception Detection model induction using GP has 3 stages:

1. Feature extraction based of linguistic cues
2. Calculate statistical summary for each cue.
3. Defining and applying GP function set on the calculated cues, with different GP parameter thus controlling induction pressure.

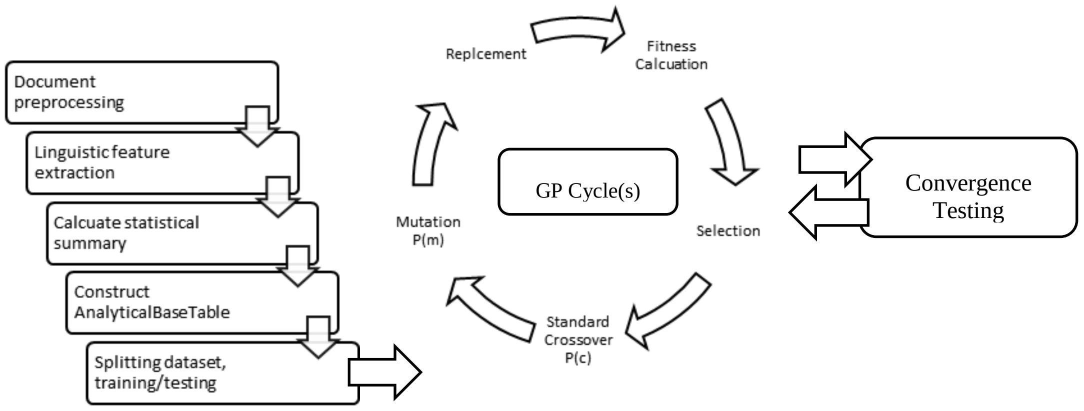


Figure-1 Deception Detection model induction using GP

Document pre-processing – In this step we perform observations loading and preliminary processing by 2 level tokenization of the textual observation into statements and vector of words, removing punctuation, symbols and any non-readable characters.

Linguistic feature extraction – In this step we perform linguistic cues calculation against both extracted vector of statements and words.

Calculating statistical summary – In this step we calculate statistical summary include: min, max, mean, median, mode and standard deviation. The reason behind this step is due randomisation step in which case we pick a randomize value within the range of given attribute.

Construct AnalyticalBaseTable (ABT) – In this step we construct final ABT aggregating both textual observation with pre-processing statement/words vector and final statistical summary.

Splitting dataset training/testing – In this step we split the entire dataset into training and testing parts, where training part is used in model induction while the testing part is used in validation and evaluation of final fitted model.

GP Cycle – In this step we perform the GP procedure against the ABT (Koza, 1992) defined by control parameters.

Convergence Testing – At the end of each GP cycle, we perform a convergence testing based for a decision if we would continue in the induction process or terminate.

4.1 Language Cues

Two classes of language Cues [5][6][7] have been used in our experiment Statistical Measurements and Readability Indices Table-1 summarize linguistic cues them:

No.	Name	Class	Computational Method
1	Number of Sentences	Statistical Measurements	Calculate number of sentences based on splitting of observation according to statement delimiter punctuations.
2	Number of Complex Words		Calculate complex words are defined to have 3 or more syllables.
3	Number of Words		Calculate number of words based on whitespace characters.
4	Number of Characters		Calculate the total number of characters in each observation.
5	Number of Syllables		Calculate the total number of syllables in each observation.
6	Average Word Length		Calculating average word length per observation
7	Automated Readability Index	Readability Indices	Readability index based on the following: $4.71\left(\frac{\text{characters}}{\text{words}}\right)+0.5\left(\frac{\text{words}}{\text{sentences}}\right)-21.43$
8	Gunning Fog Index		Readability index based on the following: $0.4\left[\left(\frac{\text{words}}{\text{sentences}}\right)+100\left(\frac{\text{complex words}}{\text{words}}\right)\right]$ Where, the complex words are those with three or more syllables
9	Coleman-Liau Index		Readability index based on the following: $CLI = 0.0588L - 0.296S - 15.8$ L is the average number of letters per 100 words and S is the average number of sentences per 100 words.

10	**Flesch-Kincaid Index**		**Readability index based on the following:** $0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59$
11	SMOG		Readability index based on the following: $\text{grade} = 1.0430\sqrt{\text{number of polysyllables} \times \frac{30}{\text{number of sentences}}} + 3.1291$ Where polysyllables are those with three or more syllables

Table-1 Summarize linguistic cues

4.2 Chromosome encoding

We maintain the encoding of chromosome to be in a binary tree structure form in GP where each chromosome represents one model [1], such that traversing it would lead to a prediction given an observation and its associated linguistic cues. Figure-2 display sample of tree model.

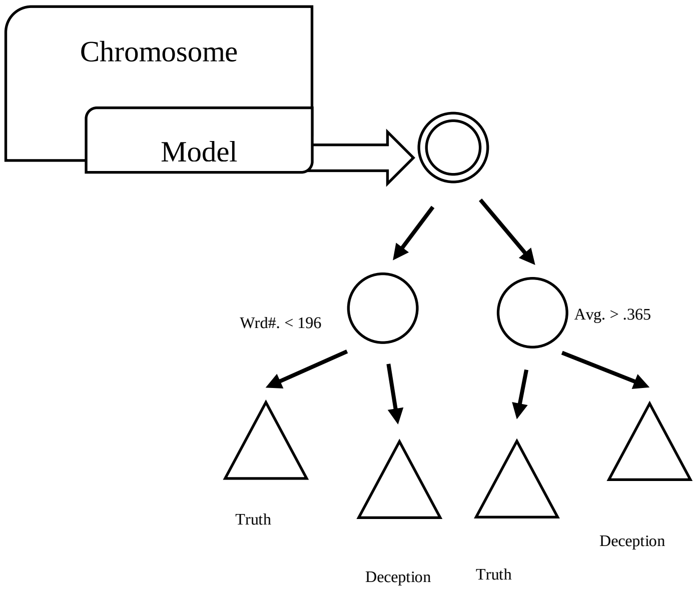


Figure-2 Sample of tree model showing the relation between chromosome and inductive model

4.3 Function set

As part of model induction we define four different kind of model tree nodes as shown in Table-2 Summary of node types.

No.	Node type	Class	Description
1	Inequality arithmetic operators node	Internal	Include <, >, >=, <=, <>, ==
2	In-range operator node		Operand belong to [n, m] where n, m are numbers with range of the operand attribute compared against.
3	Deception leaf node	Terminal	Decision leaf node of deceptive observation
4	Truthfulness leaf node		Decision leaf node of truth observation

Table-2 Summary of node types.

4.4 Stopping criterial

Three different classes of stopping criterial is defined, where GP algorithm iteration is stopped if any is reached.

1. Euclidean distance between two successive populations
2. Target accuracy
3. Max iteration count

In the first case, we defines two vectors representing the fitness of individuals in two successive populations and then calculate the Euclidean distance between each. Thus, we can terminate GP iteration if the distance is less than a given user input.

$$E(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

Where x, y are population fitness vector. In the second case, we terminate the execution if the maximum individual fitness reached target accuracy. In the final case, we are iterating till we hit defined max iteration number provide by the user.

5 Experiment Design

5.1 Corpus

We perform our experiment against labelled *Deceptive Opinion Spam Corpus v1.4* [4], which consists of the following datasets

- 400 **truthful positive** reviews from TripAdvisor
- 400 **deceptive positive** reviews from Mechanical Turk
- 400 **truthful negative** reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp
- 400 **deceptive negative** reviews from Mechanical Turk

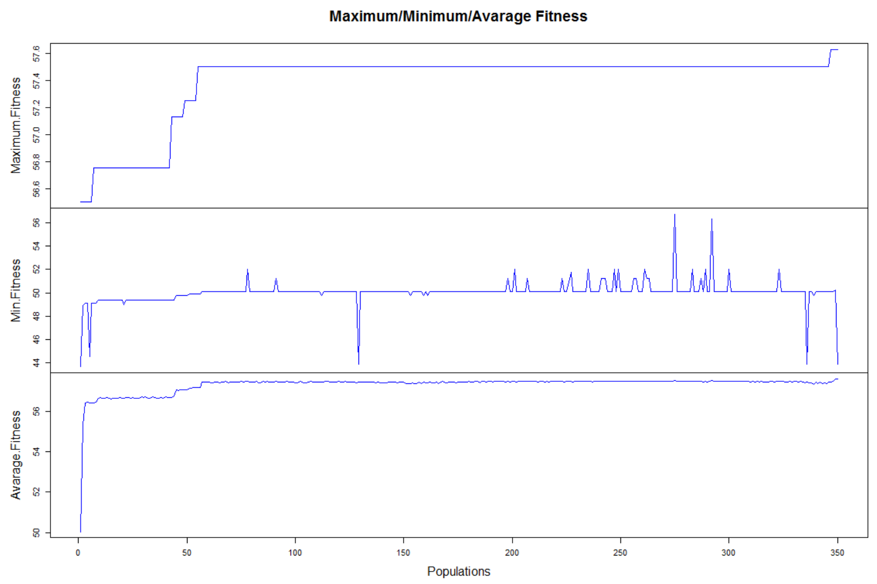
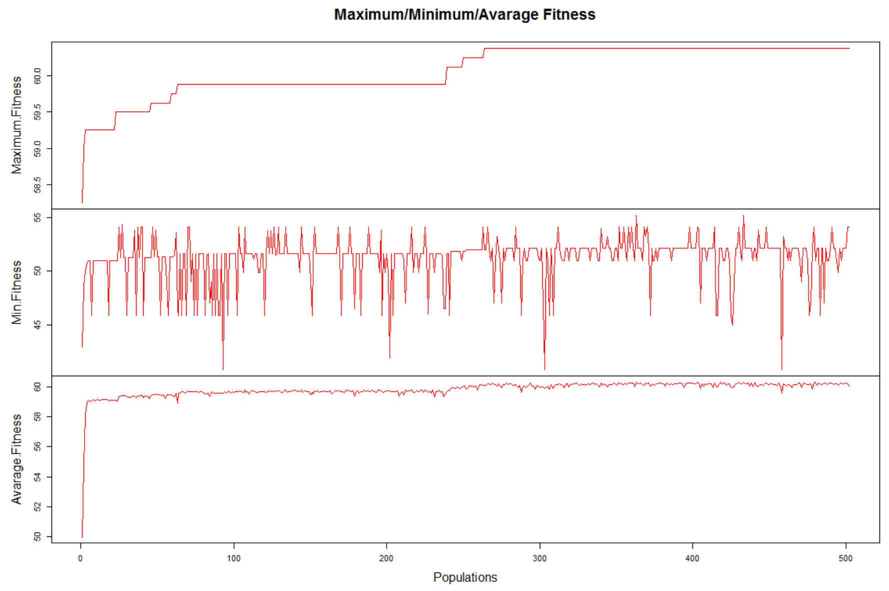
5.2 Control Parameters

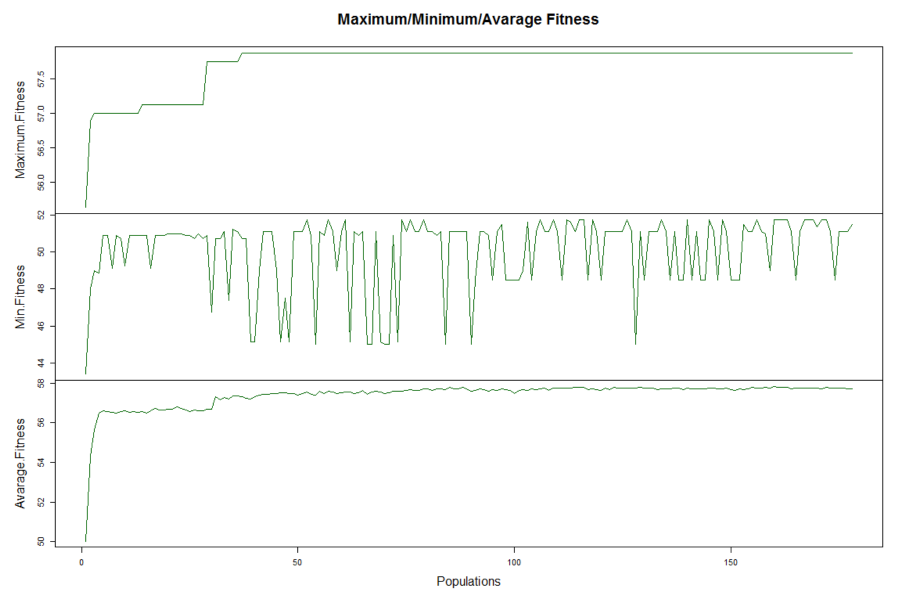
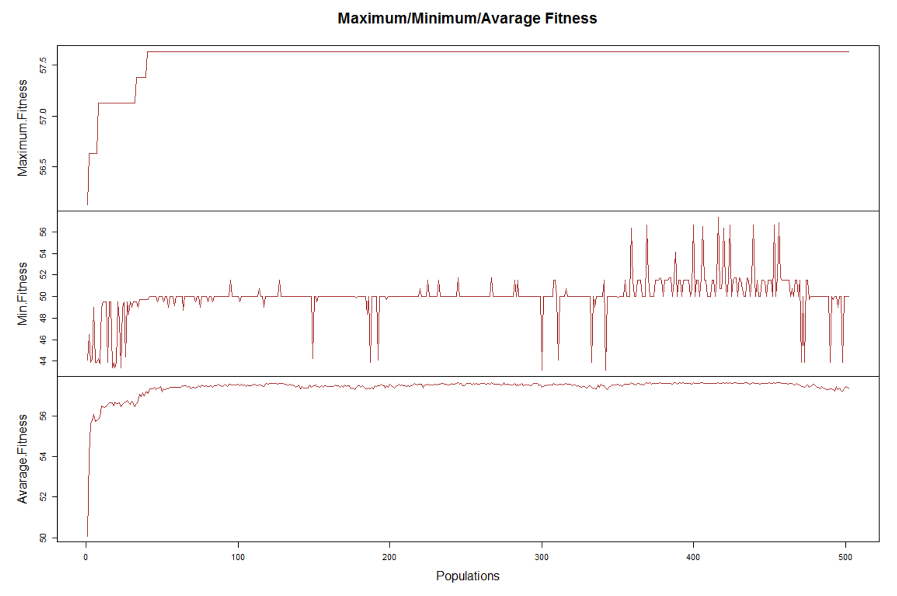
Table-3 Defines the list of control parameter used to control the GP execution and selection pressure.

No.	**Parameter**	**Description**
1	Max Iteration	Define maximum number of GP try-out iterations cycles before reaching to target optimum.
2	Target Accuracy	Define the desired accuracy that GP should stop if reached.
3	Threshold distance	Define the minimum Euclidean distance between two successive populations.
4	Population size	Define total number of chromosome that constitute a given population.
5	Max/Min model depth	Define maximum and minimum tree model depth.
6	Branch-To-Bound ratio	Define a bias probability, either positive or negative, if the next randomised node in the model is to be an internal or leaf. For example, a value of .75 will tend to generate randomly internal node by 75% than leaf.
7	Crossover probability P_c	Define standard crossover probability
8	Mutation Probability P_m	Define mutation probability

Table-3 List of control parameters

Table-4 Display different runs with convergence rate of **maximum**, **minimum** and **average** individual fitness.

Parameters	**Performance Plot**
Experiment #1 **Max Iterations: 500 Cycles** **Target Accuracy: 80%** **Threshold Distance: 0.005** **Population Size: 1000** **Min Depth: 5 Levels** **Max Depth: 10 Levels** **Branch-BoundRatio: 85%** **Mutation Probability: 0.1** **Crossover Probability: 1.0**	
Experiment #2 **Max Iterations: 500 Cycles** **Target Accuracy: 80%** **Threshold Distance: 0.005** **Population Size: 500** **Min Depth: 5 Levels** **Max Depth: 12 Levels** **Branch-BoundRatio: 65%** **Mutation Probability: 0.1** **Crossover Probability: 0.9**	

Experiment #3 **Max Iterations: 500 Cycles** **Target Accuracy: 80%** **Threshold Distance: 0.005** **Population Size: 500** **Min Depth: 7 Levels** **Max Depth: 12 Levels** **Branch-BoundRatio: 95%** **Mutation Probability: 0.1** **Crossover Probability: 1.0**	
Experiment #4 **Max Iterations: 500 Cycles** **Target Accuracy: 80%** **Threshold Distance: 0.005** **Population Size: 500** **Min Depth: 4 Levels** **Max Depth: 7 Levels** **Branch-BoundRatio: 95%** **Mutation Probability: 0.1** **Crossover Probability: 1.0**	

5.3 Experiments Observations

As we can see from the above different 4 experiments, the upper accuracy percentage reached is 60% which is slightly better higher than 50/50 random coin tossing strategy. Also, on average all the experiments converges fast in the beginning and slowly improve in later stages. Which is related to the fact that we are using variation of readability indices as our main linguistic cues that differ in scaling value but preserve the semantic complexity of readability nature of the text.

Introducing a branch-to-bound probability add an improvement in the tree depth with, in which case when we have branch-to-bound value relativity small value we are ending up with smaller tree depth i.e. more generalized and thus high accuracy rate. Whereas when we have high branching value we end up with deep trees and over fitting our training dataset.

Final observation we need to stress upon, fluctuations of the minimum fitness does not affect the overall average performance of the GP. This is due to the adaptiveability nature of GP.

6 Conclusion and future work

In this work, we experiment with deception detection model induction using GP, a natural computation approach. Where we have reached 60% accuracy rate, more than a 50/50 random chance, with basic statistical summary and set of readability indices. GP approach proved to adapt to induct deception detection model without concerning with the underlying domain (hotel reviews in our case), which in turn oppose to statistical methods where the model is based on the hypotheses derived by the domain.

For future work, other types of linguistic cues could be used, in which case it would dramatically increase the model search space. Along with additional functions set while experimenting different input values.

7 References

[1] Koza, J.R. (1990). Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems, Stanford University Computer Science Department technical report STAN-CS-90-1314. A thorough report, possibly used as a draft to his 1992 book.

[2] Atkinson, J., Mellish, C. and Aitken, S., “Combining Information Extraction with Genetic Algorithms for Text Mining”, IEEE Intelligent Systems 19(3), pp22-30, 2004. PDF

[3] Sean L. H, Kevin C. M, Mary B. Burns, Judee K. B, William F. Felix; “Identification of fraudulent financial statements using linguistic credibility analysis”; (2013)

[4] Deceptive Opinion Spam Corpus v1.4 http://myleott.com/op_spam/

[5] Senter, R.J.; Smith, E.A. (November 1967). "Automated Readability Index.". Wright-Patterson Air Force Base. p. iii. AMRL-TR-6620. Retrieved 2012-03-18.

[6] Coleman, M.; and Liau, T. L. (1975); A computer readability formula designed for machine scoring, Journal of Applied Psychology, Vol. 60, pp. 283–284

[7] Kincaid, J.P., Fishburne, R.P., Rogers, R.L., & Chissom, B.S. (1975). Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for Navy enlisted personnel. Research Branch Report 8–75. Chief of Naval Technical Training: Naval Air Station Memphis.

[8] Newman ML, Pennebaker JW, Berry DS, Richards JM; Lying words: predicting deception from linguistic styles (2003)

8 Appendix – Project Technical Aspects

In this appendix we are going to discuss the technical aspect of the project along with a high-level design overview of the development effort.

8.1 Development environment

- **Programming Language**: Java SE version 8 (Core development) / R (for result plot)
- **IDE**: eclipse Luna SR2 / https://eclipse.org/
- **GP Library**: None – GP algorithm is totally implemented from scratch, which give more flexibility specially integration with the linguistic cues computation.
- **NLP Library**: LingPipe which is a toolkit used for processing text using computational linguistics / lingpipe-4.1.0.jar / http://alias-i.com/lingpipe/

8.2 Main Subsystem

In this subsection we discuss the main subsystem (as appears in the Java packages names) used in conducting the experiment.

- TM (Test Mining) ~ Contains all the linguistic computational code for the textual observations.
- GP (Genetic Programming) ~ Contains all GP related data structure and operations.
- ABT (AnalyticalBaseTable) ~ Contains table data structure for the final analytical base table that consolidates textual observation, statistical summary and linguistic cues.
- DS (Dataset) ~ Contains the dataset loader code with the pre-processing before loading the ABT.
- Common ~ Contains common functionality used by all components.
- App (Application) ~ Main entry point of the project.