

**Modeling and Verifying Automated Machine Learning Models**

**INSE- 6250 Project Report**

**Submitted To:**

Dr. Jamal Bentahar

**Submitted By:**

Sanchit Kumar (40081187)

**Date of Submission:**

16-April-2019

# Abstract

In recent time, machine learning and analytics in evolving day by day and lot of organization are deploying machine learning models in the production environment. The deployments are not only restricted to batch mode. A lot of organizations have been deploying real time processing analytics engines to provide the best out to their users. However, the verification of such automated models remains the challenge that need to be solved. As these systems provide results in real time, it’s impossible to perform the verifications manually.

This paper provides a model checking solution to the real time automated machine learning and analytics systems. Uppaal is used for modeling, verification and validation of the system. Uppaal provides a toolbox to verify real-time systems and has been successfully used in case studies of communication protocols and multimedia applications. To perform model checking we are using CTL formal language to specify properties and verify them using the model checking tool. Uppaal can perform model checking automatically by specifying the properties in the query window. It also provides the counter example in case the property is not satisfies by the model.

# 1.0 Introduction

Automated machine learning models have nowadays becoming very common as more and more organization are moving towards incorporating analytics in their system to get the value out of their data. Deploying their systems in production needs proper verification and validation to make the efforts successful. The accuracy is of utmost importance when deploying such analytics systems as they can backfire if not properly implemented and can have a huge loss for the organization.

# 2.0 Model Design

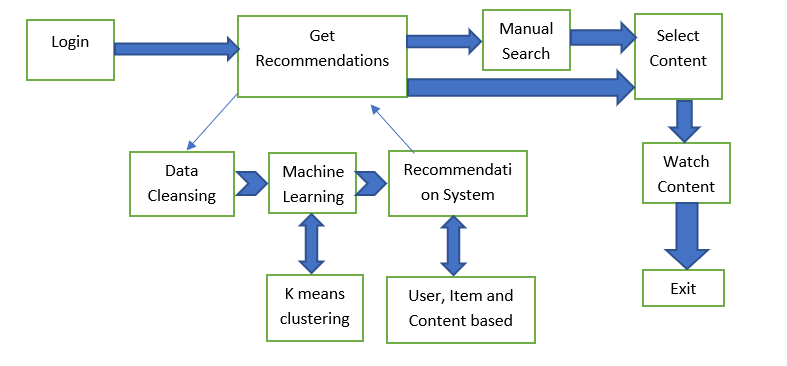


Figure 1: Block Diagram of Automated Machine Learning Model

# 3.0 Requirement Specifications

1. **Reachability Properties (Coverage)**: All the states in the system must be reachable by the users at least once at some point in time. Reachability property in a state is defined by E<>, meaning the state will be accessible in at least one path in the future.

|  |  |
| --- | --- |
| **No.** | **Property** |
| 1 | E<> f.logout |
| 2 | E<> f.login |
| 3 | E<> ( recon\_system.end\_result ) |
| 4 | E<> ( ml\_model.Deploy ) |
| 5 | E<> ( ml\_model.Model\_Selection ) |
| 6 | E<> ( f.click\_content | f.search\_content ) |
| 7 | E<> f.click\_content |
| 8 | E<> f.watch |

**Table1: Reachability Properties**

1. **Liveness Property (Transitions)**: There must be coverage of all the transitions in the system and it should not be the case that a transition is never covered in the system.

|  |  |
| --- | --- |
| **No.** | **Property** |
| 9 | A[] ( ((k.counter < k.cluster\_size) and clustering\_completed == false) or (clustering\_completed imply (k.counter >= k.cluster\_size) ) ) |
| 10 | A[] ( rm\_noise.Noise\_removed imply (rm\_noise.remove\_na\_null & rm\_noise.outliers\_removal ) ) |
| 11 | A[] ( ml\_model.Exit imply automation\_model\_completed ) |
| 12 | A[] ( ml\_model.Make\_predictions imply ( preprocessing\_done & clustering\_completed ) ) |
| 13 | A[] ( f.profile\_creation\_process imply (!f.existing\_user) ) |
| 14 | A[] ( ml\_model.Model\_Selection imply ( preprocessing\_done == true ) ) |
| 15 | A[] ( ml\_model.Deploy imply ( ml\_model.accuracy >= 90 | recon\_generated == true ) ) |
| 16 | A<> (f.recommend\_content imply ml\_model.Exit) |
| 17 | A<> (f.click\_content imply f.check\_subscription) |
| 18 | A<> (f.login imply f.recommend\_content) |

**Table 2: Liveness Properties**

1. **Safety Properties**: Safety properties covers those cases which should not occur in the system. These are the properties which checks for the illegal transitions in the model to check the non expected behaviour of the system.

|  |  |
| --- | --- |
| **No.** | **Property** |
| 19 | A[] ( f.logout imply !(f.logged\_in == false | f.existing\_user == false ) ) |
| 20 | A[] not( ml\_model.accuracy >= 90 & recon\_generated ) |
| 21 | A[] ( f.click\_content imply ( f.existing\_user & f.logged\_in & (recon\_complete | f.manual\_search) ) ) |
| 22 | A[] (f.watch imply ( f.existing\_user and f.content\_access) ) |
| 23 | A[] k.counter <= k.cluster\_size |
| 24 | A[] deadlock imply f.exit |

**Table 3: Safety Properties**

# 4.0 Model

The model constructed consists of 7 templates namely – front\_end, automation\_model, remove\_noise, data\_cleansing, k\_means\_algo, fetch\_content\_process, recommendation\_system.