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

Table of Contents

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

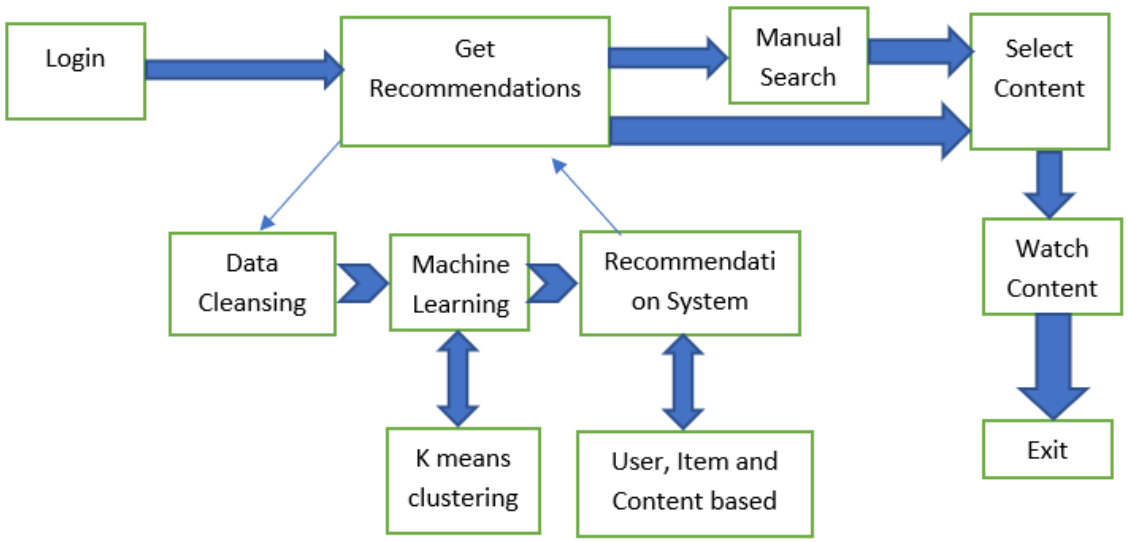

Figure 1: Block Diagram of Automated Machine Learning Model

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)
7	E<> (ml_model.Model_Selection)
8	E<> (f.click_content | f.search_content)
	E<> f.click_content
	E<> f.watch

Table1: Reachability Properties

2. **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

3. **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 in Uppaal

The model constructed consists of 7 templates namely – front_end, automation_model, remove_noise, data_cleansing, k_means_algo, fetch_content_process, recommendation_system.

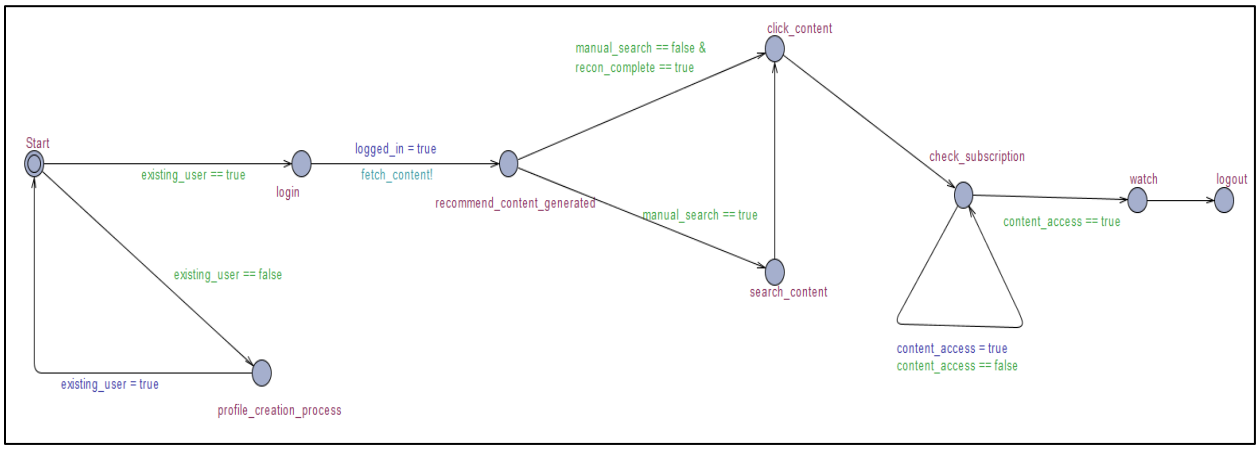

Figure 2: front_end Template

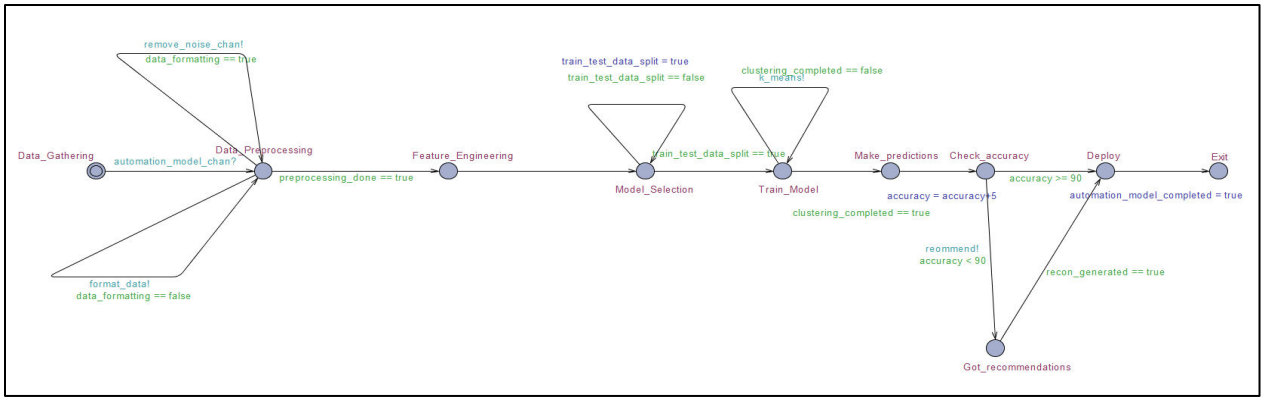

Figure 3: automation_model Template

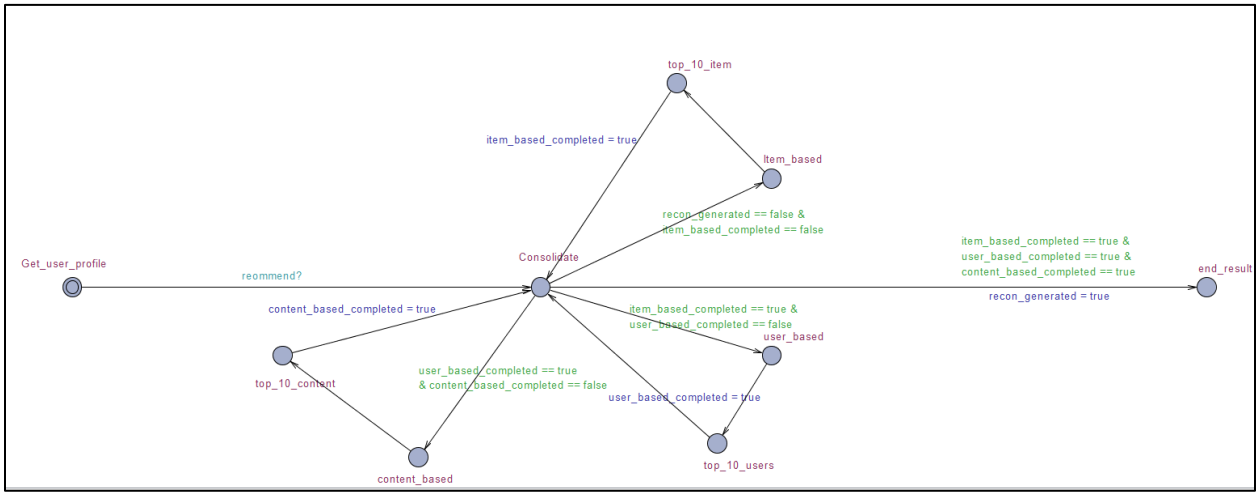

Figure 4: recommendation_system Template

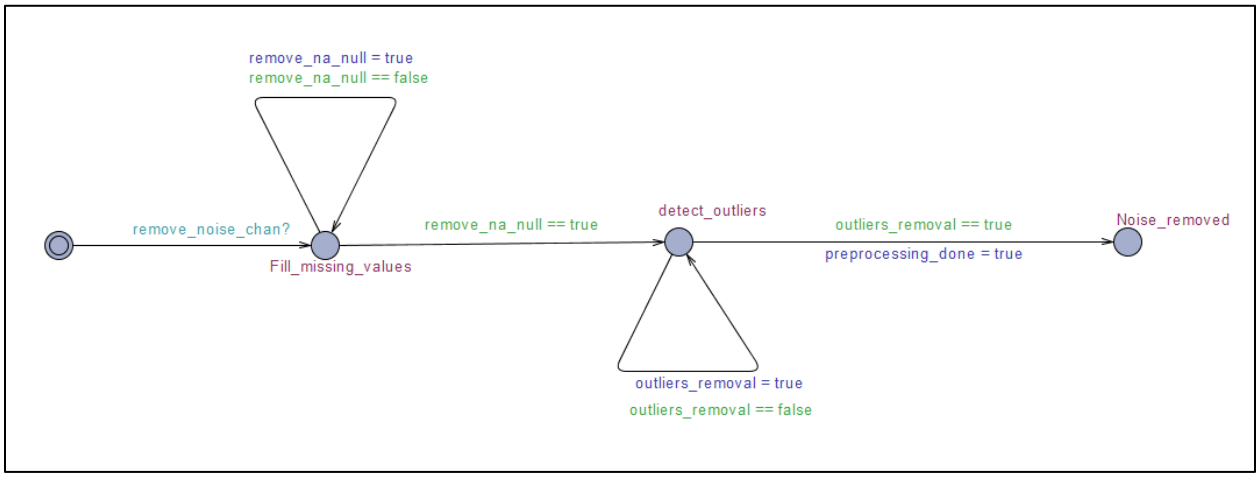

Figure 5: remove_noise Template

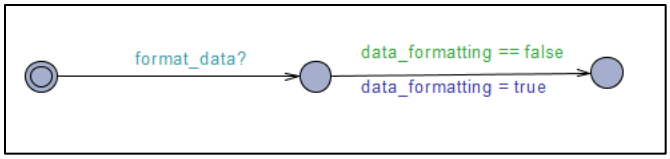

Figure 6: data_cleansing Template

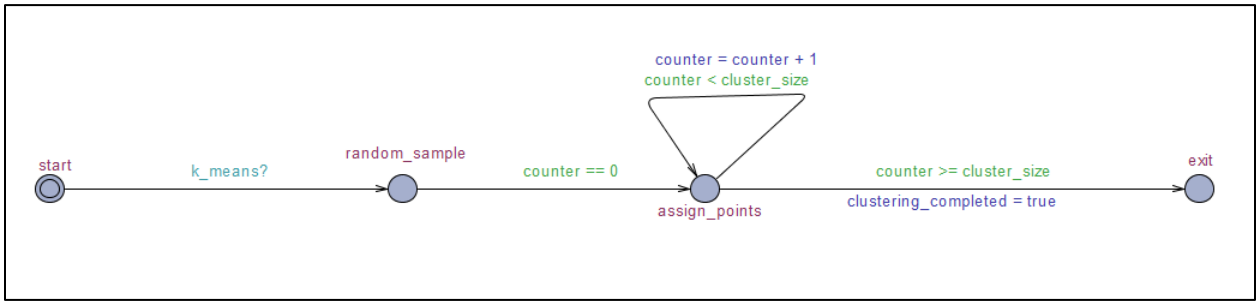

Figure 7: k_means_algo Template

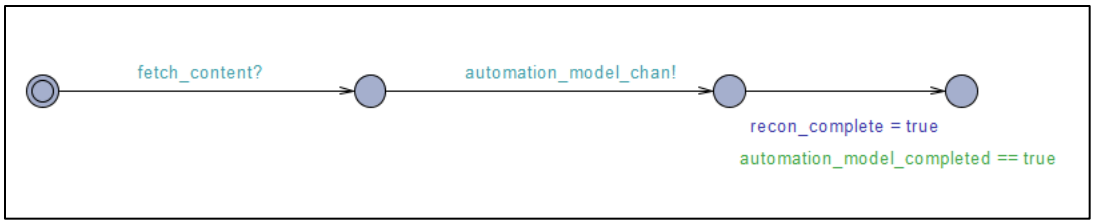

Figure 8: fetch_content_process Template

4.1 Local, Global variables and Process

4.1.1 Local Variables

Template	Variable Name	Variable Scope	Variable	Comment

			Type	
front_end	logged_in	Local	bool	Used to indicate if the user is already logged in.
front_end	manual_search	Parameter	bool	Used to indicate if the user wants to search content manually or not.
front_end	existing_user	Parameter	bool	Used to indicate if the user is new or existing one.
front_end	create_profile	Local	bool	Used to indicate if the new user has created a profile
front_end	content_access	Local	bool	Used to indicate of the user have access to the content
automation_model	train_test_data_split	Local	bool	Used to indicate if the train test split happened or not
automation_model	accuracy	Parameter	Int	Used to indicate the accuracy of the model
remove_noise	remove_na_null	Local	bool	Used to indicate if the na or null values are removed or not.
remove_noise	outliers_removal	Local	bool	Used to indicate if the outliers are removed from the model or not.
k_means_algo	counter	Local	Int	Used to indicate then number of classes in the data.
k_means_algo	cluster_size	Parameter	Int	Used to indicate the cluster_size
recommendation_system	item_based_completed	Parameter	bool	Used to indicate if the item based collaborative filtering is completed.
recommendation_system	user_based_completed	Parameter	bool	Used to indicate if the user_based_collaborative filtering is completed.
recommendation_system	content_based_completed	Parameter	bool	Used to indicate if the content_based filtering is completed.

4.1.2 Global Variables

Variable Name	Scope	Type	Comment
recon_complete	Global	bool	Used to indicate if the recommendation process is completed
automation_model_completed	Global	bool	Used to indicate if the automation model has finished computation
user_profile	Global	bool	Used to indicate if the user profile exist or not
preprocessing_done	Global	bool	Used to indicate if the preprocessing operations are completed
data_formatting	Global	bool	Used to indicate if data formatting

			operations are completed
clustering_completed	Global	bool	Used to indicate if the clustering results are completed
recon_generated	Global	bool	Used to indicate if the recommendation are generated
remove_noise_chan	Global	Channel	Used to initiate noise removal process
format_data	Global	Channel	Used to initiate data format process
k_means	Global	Channel	Used to initiate the kmeans process
fetch_content	Global	Channel	Used to initiate the fetch content process
automation_model_chan	Global	Channel	Used to initiate the automation model
reommend	Global	Channel	Used to initiate the recommendation process

6.0 References

1. http://www.uppaal.org/
2. http://www.di.unipi.it/~maggiolo/Lucidi_TA/VerifyingTA-Uppaal.pdf
3. https://www.cs.mcgill.ca/~esyria/publications/UPPAAL.ppsx
4. https://www.it.uu.se/research/group/darts/uppaal/small_tutorial.pdf
5. http://ceur-ws.org/Vol-1128/paper5.pdf