ENVISAGE

ENhanced VIrtual learning Spaces using Applied Gaming in Education

H2020-ICT-24-2016

D3.1 Preliminary Predictive Analytics and Course Adaptation Methods

Dissemination level:	Public (PU)
Contractual data of delivery:	Month 7, April, 2017
Actual date of delivery:	Month 11, August, 2017
Work package:	WP3 - Deep game analytics for predicting learning behaviors and delivering appropriate learning content
Task:	T3.1 – Learners' future behavior prediction
Type:	Demonstrator
Approval status:	Final
Version:	0.6
Number of pages:	25
Filename:	D3.1-PreliminaryPredictiveAnalyticsCourseAdaptationMethods.pdf

Abstract

We review the state of the art in game and learning analytics and compare and integrate these findings with the insights gained from the prior deliverables in the ENVISAGE project. We identify three possible overarching approaches that can be explored in the ENVISAGE project: clustering, prediction, and simulation of students and/or their behavior. Not all of these may be attainable for ENVISAGE, but at the current stage presents themselves as options that should all be pursued and investigated. Methods for each approach are described and outlined in the deliverable and an initial framework is implemented as an on-line Deep Analytics Learning Service that the other components developed in the ENVISAGE project can interface with.

co-funded by the European Union

Copyright

History

Version	Date	Reason	Revised by
0.1 (alpha)	2017-06-09	Table of contents circulated to the consortium for approval	Christoffer Holmgård, Daniel Mercieca, and Georgios Yannakakis
0.2	2017-07-07	Elaborated version circulated for feedback	Christoffer Holmgård
0.3 (beta)	2017-07-14	Beta version circulated for internal review	Christoffer Holmgård and Georgios Yannakakis
0.4	2017-07-28	Version adressing the comments from the internal review	Christoffer Holmgård and Daniel Mercieca
0.5	2017-08-07	Pre-final version provided to the coordinator for submission	Christoffer Holmgård
0.6 (final)	2017-08-09	Final version proof edited by the coordinator	Spiros Nikolopoulos

Author list

Organization	Name	Contact Information
UoM	Christoffer Holmgård	holmgard@gmail.com
UoM	Georgios Yannakakis	georgios.yannakakis@um.edu.mt
UoM	Daniel Mercieca	arakanis@gmail.com
CERTH	Spiros Nikolopoulos	nikolopo@iti.gr

Executive Summary

This deliverable, constituting the first part of WP3, addresses preliminary analytics and course material adaptation. It reviews the state of the art in game and learning analytics and compares and integrates these findings with the insights gained from the prior deliverables in the ENVISAGE project. We identify three possible overarching approaches that can be explored in the ENVISAGE project: unsupervised modeling, supervised modeling, and generative modeling. Respectively, these three approaches can be thought of as representing the clustering, prediction, and simulation of students and/or their behavior. Methods for each approach are described and outlined in the deliverable and an initial framework is implemented as an on-line Deep Analytics Learning Service that the other components developed in the ENVISAGE project can interface with. An initial evaluation of the pros, cons, and feasibilities of the three approaches is provided and a way ahead for the remainder of the project is outlined.

Abbreviations and Acronyms

ML	Machine Learning
EDM	Educational Data Mining
LA	Learning Analytics

Contents

List of Figures

List of Tables

1 Introduction

This deliverable, constituting the first part of WP3, addresses preliminary analytics and course material adaptation. It takes the form of a demonstrator project, designed and built on the basis of insights gained from a review of existing approaches for deep analytics for learning as well as games. For this deliverable, we review the state of the art in game and learning analytics and compare and integrate these findings with the insights gained from the prior deliverables in the ENVISAGE project. We identify three possible overarching approaches that can be explored in the ENVISAGE project: unsupervised modeling, supervised modeling, and generative modeling. Respectively, these three approaches can be thought of as representing the clustering, prediction, and simulation of students and/or their behavior. Not all of these may be attainable for ENVISAGE, but at the current stage presents themselves as options that should all be pursued and investigated in order to find the solution that is most useful to teachers and other users of the ENVISAGE authoring tools. Methods for each approach are described and outlined in the deliverable and an initial framework is implemented as an on-line Deep Analytics Learning Service that the other components developed in the ENVISAGE project can interface with. An initial evaluation of the pros, cons, and feasibilities of the three approaches is provided and a way ahead for the remainder of the project is outlined. In the following section, we start by reviewing the state of the art in the field of Game Analytics.

2 Machine Learning for Game Analytics

Given that the ENVISAGE project takes as its premise that methods from Game Analytics for commercial entertainment games may successfully be transferred into the space of digital learning environments, this section investigates the state of the art in machine learning for Game Analytics. With an outset in a recent comprehensive volume compiled by El-Nasr et al. [8], we provide a brief overview over current approaches and state of the art in Game Analytics in the space of the commercial game industry.

2.1 Game Data Mining

One approach to (predictive) game analytics in the commercial game industry is to apply principles of the well-defined *Cross-Industry Standard Process for Data Mining* (CRISP-DM) [8]. CRISP-DM defines six steps of data mining for analytical purposes that are general enough to be translated to any data producing industry, including digital games. These six steps include:

1. Business/research understanding.
2. Data understanding.
3. Data preparation.
4. Modeling.
5. Evaluation.
6. Deployment.

The steps can be retraced iteratively in a loop to refine the data-mining process. Below we visit each of them in relation to the ENVISAGE project.

2.1.1 Business/Research Understanding

This first step relates to identifying the objectives and requirements of the general project and translating these into a data mining problem, or in ENVISAGE terminology, a Learning Analytics problem. In the case of ENVISAGE, this represents a large part of the work that must be undertaken: understanding what constitutes the theoretical and practical problems of understanding student behavior and learning in digital learning environments. For the work presented in this deliverable, the prior knowledge gained from deliverables D1.1, D1.2, D2.1, and D2.2 all provide the nearest source of information for this step though further informed, naturally, from the original sources and data used for these deliverables. In essence, the step consists of forming a clear understanding of how teachers use digital learning environments with students, how students interact with them in context, and what theoretical prior knowledge can be leveraged to structure and make sense of observations of student behavior and development. The outcome of this process that most directly has fed into the development of analytics and course adaptation methods is the identification and definition of the base metrics that are also used for visualizations in D2.3:

- Class Profile

- Levels of Proficiency
- Time on Task
- Time to Completion
- Travel Path

Predicting and optimizing for these metrics, or indicators of these metrics, can be seen as the end operational goal of the analytics and adaptation methods developed through ENVISAGE.

2.1.2 Data Understanding

This second step deals more concretely with structuring and exploring the data that is gathered for the modeling task. For the case of the ENVISAGE project, this involves observing data collected from the digital learning environments and relating this data to the objectives identified in D1.1 [16]. As the data is being generated from tracking points defined and inserted into the digital learning environments as part of the ENVISAGE project, described in D4.2 [17], this part of the process is relatively transparent for the ENVISAGE project since contains both the data generation, analysis, and modeling parts of the process. Data understanding also includes early exploratory data analysis where the raw data is examined to inform which modeling techniques might be appropriate.

2.1.3 Data Preparation

Data preparation pertains to the manipulation of data into a form where it is ready for being entered into the modeling step of the process. This may involve cleaning data by removing collected values that represent test data or removing sessions that were not completed or otherwise unused or unwanted. It may also involve calculating any values that may be statically derived from the collected data and will be used and reused during the modeling step. In the case of ENVISAGE, there is a need for cleaning data from test data collected during the development process, and there is a need for calculating certain values derived from raw data, such as session lengths and time stamps between events collected in the data set. Overall, however, the data preparation step is relatively manageable for ENVISAGE, again due to the highly integrated nature of the project, which may be atypical for the case of data mining.

2.1.4 Modeling

Modeling refers to the step where the preparations of the three previous steps are leveraged to select appropriate data mining/modeling techniques and apply these to the prepared data materials. A wide range of different techniques and algorithms may be employed, depending on the investigative interest. Each of these must be run and calibrated in different ways, depending on the chosen method.

In the case of ENVISAGE, the modeling techniques are chosen based on the interests determined chiefly in D1.1 [16] and D1.2 [12]. Whenever a modeling process is completed, the resulting model will typically be stored for application to new results or prediction of new values based on updated data sets. In the case of ENVISAGE, the generated models will be stored and used to cluster students and classes, and to predict the performance and behavior of individual students or groups of students, in accordance with the interests identified in other deliverables. As new data becomes available, the modeling process is usually reiterated, either automatically or with manual intervention to update and inform the model with the newest observations. Over time, this typically leads to better performing models with more accurate results.

2.1.5 Evaluation

After models have been created and stored they must be evaluated, not only in terms of their internal, technical metrics of performance, but also in the more general sense of whether the models are reliable and valid. These terms pertain to, respectively, whether the models measure what they model consistently across the cases that they will be applied to and whether they in fact model what they are intended to. The first question may be answered by applying the models to new data and ensuring that the results derived conform to expectations, for instance checking that cases that are expected to yield similar or different results in fact do so. The second question may be more difficult and can typically only be addressed in collaboration with subject matter experts and end users. The divisions or predictions obtained from the models must be testing against the theoretical background from which they were derived, external ground truth if any is available, and the evaluation of the final users of the models. In the case of ENVISAGE, validation of the models will be highly dependent on the collaboration and feedback from pedagogical experts and teachers, represented in the consortium by the EA partner. This evaluation will be possible when the models are deployed in the actual interface in conjunction with the authoring tool.

2.1.6 Deployment

The final step in CRISP-DM is the deployment of the developed models. Deploying models means not only integrating them into the final technical solution for which they are meant (in the case of ENVISAGE as support to authoring tool users), it also means communicating the use and meaning of the model to the end user stakeholders (in the case of ENVISAGE, teachers). Successful communication to non-technical stakeholders typically means presenting results in an intuitive user interface, supported by explanatory and training materials to support the user in cases of doubt.

2.1.7 Adapting the CRISP-DM process to ENVISAGE

In order to support predictive analytics and course adaption in the authoring tool, the method developed for the ENVISAGE project must encode all of these steps into an automated process before presenting it to the user of the authoring tool. While the predictive services being deployed for the authoring tool may be updated in collaboration with end users, the developers of the solution are not directly available to teachers or other authoring tool users on a day-to-day basis. This, in turn, makes it imperative that the analytics developed for ENVISAGE are general enough to cover as many potential users of teachers in with specific pedagogical agendas, and with needs for insights and adaptation that cannot be fully predicted at the time of implementation. This issue is further addressed in Section 4.

In the following subsections, we proceed to describe the technical aspects of the CRISP-DM process as applied in game analytics in greater detail. We define the terms and relate them to the ENVISAGE project for later use when we describe the modeling approaches chosen for the project.

2.2 Metrics and Telemetry

Metrics define higher-level constructs that say something of interest about the player base of a particular game. They are built from measurements typically obtained through telemetry: the game's source code is augmented with tracking points that are triggered intermittently or by player actions. These triggers produce events that are transmitted via the Internet to a server with the game developer. In the commercial game industry, metrics are usually defined iteratively during the development process. Telemetric trigger points are established early on in response to design intuitions or experience on the part of designers and analyst. These are then amalgamated into metrics that are made available through various interfaces. Eventually, as the game develops and nears completion, new metrics are defined from the available telemetric events, and more trigger points are added to generate new events, in an iterative design loop. This underscores that the questions that are of interest to the designers and developers of a game manifest themselves in concert with the game's development. The questions that are of interest at the beginning of the development process are typically different from the ones at the end of the process.

This is relevant to the ENVISAGE project, where the labs that are provided to authoring tool users are reminiscent of commercial games near the final stages of development. The fundamental rules of the labs are well-defined: pedagogical goals and the internal logic of the labs have already been constructed with particular learning goals in mind, and the object of the authoring tool user is to change the details of these labs to best support the learning of the particular students that are interacting with the strongly defined lab. This also means that the pre-defined metrics and analytics included in the predictive analytics and course adaptation methods must be pre-defined, as it is not realistic that the end user of the authoring tool would be able to define new tracking points or implement additions to the existing telemetry. This, however, presents us with a challenge: How may we, without knowing the pedagogical approach or needs of a specific teacher, best support this teacher in understanding how their students' are using the virtual lab and how they might respond to changes to the virtual lab. We address this challenge later in Section 4.

2.3 Feature Extraction

Once metrics are constructed from raw events obtained from telemetry data, they may be further refined through a process of feature extraction. Feature extraction can take on several forms where the simplest one is the approach of feature selection: the most important metric for an analytic goal are selected in order to only enter relevant metrics into an analysis. ENVISAGE makes extensive use of this approach in the shallow analytics part of the project, where the pre-defined metrics, decided upon from pedagogical expert knowledge, are aggregated and visualized to end users of the visualizations and authoring tool.

A more advanced form of feature extraction is the derivation of new features from either raw telemetric data or the metrics informed by expert knowledge.

2.4 Modeling

Modeling, the fourth step of the CRISP-DM process, defines the analyzing the collected raw data, metrics, and extracted features to create insights that can be reported as insights upon which actions can be taken. While the commercial game industry uses simple aggregation and reporting of metrics for most purposes, as ENVISAGE does in shallow analytics, game analytics is increasingly including more advanced player modeling as well.

Player modeling involves using collected raw event data and metrics to construct more elaborate groupings of player based on combinations of their background information and behavior, such as e.g. personas, or predicting future player behavior from their past actions.

Player modeling is typically divided into model-based and model-free player categories [18]. Model-based player modeling is a top-down process in the sense that theoretical models about players are used to inform the modeling process and structure conclusions drawn from the data. Model-free player modeling is a top-down process where no a priori theoretical model is used to make sense of the data, but rather computational techniques are used to extract groupings and relations found in the observed data. Post-hoc, theoretical insights may be used to interpret the found models. The two may also be combined to accomplish synthetic player modeling, where theoretical knowledge is used to drive part of the modeling process, such as e.g. for feature extraction from raw data (as in the case of predefined metrics), or to limit modeling outcomes to specific categories.

2.5 Spatial and Visual Analytics

A key trend in game analytics is the spatial and visual presentation of models using visual representation. For summary and aggregate reporting of simple metrics, this is typically accomplished using dashboard like interfaces that present information in bar-charts, line-graphs, pie-charts, and flow-charts like ENVISAGE does for shallow analytics, as described in [10]. This also applies when these metrics are used to group users or to predict future values.

For more advanced information, however, such as the reporting of player movements or actions within the context of the game, game analytics typically turns to visualization in context via the use of *heatmaps* [8]. Heatmaps are color-coded overlays on top of screenshots or 3D-models of games, typically levels, that show actions frequencies or intensities across the space. While heat-maps have the advantage of presenting rich information in a highly contextualized way, they have a high implementation cost, as they must be added to the actual engine responsible for the rendering of the game, or must be mapped to screenshots for that engine. This is typically feasible in commercial game productions where a single product is being developed, and a development team is available to create a bespoke analytics solution for this game. Recently, game engines - including the Unity game engine being used for ENVISAGE - have started shipping with support for heat-mapping built-in. While still requiring some effort on the part of the developer this may reduce the implementation cost of spatial and visual analytics.

In the context of ENVISAGE heat-maps could be used to represent actual learner behavior in each of the virtual labs, at the aggregate or individual level. However, rendering such visualizations intelligible to non-expert users could be challenging and the strategy should be evaluated with representative authoring tool users before being implemented.

2.6 Summary

In this section we briefly reviewed the state of the art in commercial game analytics and related these to the ENVISAGE project. We showed how practices in game analytics follow the more general cross-industry CRISP-DM model for data mining and how player modeling can be implemented to provide analytical insights within this model. We also showed how different kinds of machine learning find application in player modeling and how this information is typically communicated back to users using contextualizing visualization strategies such as e.g. heatmaps. In the following section we visit current practices and discussions in the educational data mining and learning analytics communities, relate these to game analytics in general, and provide examples of the use of game analytics and machine learning for adaptive learning games.

3 Educational Data Mining versus Learning Analytics

In [1], Baker and Inventado provide a useful overview of the related research areas of Learning Analytics (LA) and Educational Data Mining (EDM).

Citing [14], they propose that the two are related, but take different approaches to understanding the learning process and also take different approaches to leveraging data:

> In that work, it was argued that there are five key areas of difference between the communities, including a preference for automated paradigms of data analysis (EDM) versus making human judgment central (LA), a reductionist focus (EDM) versus a holistic focus (LA), and a comparatively greater focus on automated adaptation (EDM) versus supporting human intervention (LA).

In short, EDM has a greater focus on automation and data oriented techniques, where learning analytics has a greater focus on informing human analysts and decision makers, but less of an interest in encoding intervention theories or invention strategies directly into the developed solution.

This is highly relevant to the ENVISAGE project, since the solutions developed for ENVISAGE in essence attempt to take cues from both these research directions: The shallow analytics implemented aim to inform and support the exploratory process of the teacher as they are using the authoring tool. The deep analytics, on the other hand, attempt to automate and predict behavior, but rather than making changes directly, the outcome of these models are still fed back to human analyst who acts upon this information using the authoring tool. As such, ENVISAGE can be seen as bridging these two approaches.

Since this deliverable is concerned with the methods implemented for deep learning analytics, we will briefly review techniques for Educational Data Mining, rather than Learning Analytics, and relate these to the objectives of ENVISAGE.

3.1 Educational Data Mining Techniques

Baker and Inventado [1] divide the methods used in Educational Data mining into four overall categories:

- Prediction Models
- Structure Discovery
- Relationship Mining
- Discovery with Models.

3.1.1 Prediction Models

Prediction refers to both classical statistical tools such as classification and regression, but also more recent approaches such as e.g. support vector machines, neural networks, or other techniques that are capable of being trained to map input values to outcome values. This corresponds to the machine learning concept of supervised learning. Baker and Inventado note that supervised learning may be useful in predicting a student's future knowledge or performance on a task, which would also be a typical application in the ENVISAGE project.

3.1.2 Structure Discovery

Baker and Inventado mention clustering, factor analysis, and domain structure discovery as typical methods that can be applied to enable structure discovery. Structure discovery deals with finding patterns in and divisions of data without having externally defined outcome values, in contrast to what is available to prediction models. As such, the category corresponds closely to what in machine learning is typically called unsupervised learning. In the ENVISAGE project, clustering would be a relevant method, as the project already has a number of theoretically motivated groupings of the students that structure discovery methods such as clustering could help find in the collected data sets.

3.1.3 Relationship Mining

Relationship mining is described as a category of methods that have the goal of finding relationships between variables in a dataset, but does not try to predict values or partition the data-set. This includes classical approaches such as correlation analysis as well as newer approaches such as sequence pattern mining. Relationship mining might be relevant as an analytical tool where the results are presented directly to the user as part of the shallow analytics in ENVISAGE. Additionally, relationship indicators such as correlation coefficients are necessary supporting tools when doing exploratory data analysis for building more complex deep analytics models.

3.1.4 Discovery with Models

An emerging approach in Educational Data Mining is discovery with models [1]. The concept refers to the application of pre-developed models through clustering, prediction, or other methods such as knowledge engineering via e.g. rule specification. The existing models are used on new data-sets and the results analyzed or fed in as secondary components to other models. The approach mirrors tendencies in player modeling from game analytics, where complex models of player behavior may be used to characterize players, discover tendencies in games, and may be used as input to other models or analyses [18] This approach is relevant to the ENVISAGE project as the successful modeling of learning behavior might require the initial grouping of students based on their behavior and a subsequent prediction of their scores or behaviors inside the digital learning environments.

4 Goals for Prediction and Adaptivity in ENVISAGE

In this section we relate the methods and approaches identified in game analytics and educational data mining to the needs of the ENVISAGE project, and identify a way ahead toward exploring and identifying which approaches are useful for ENVISAGE.

A challenge for the ENVISAGE project, as noted above in Section 2.2, is that teachers may use the implemented analytics solutions for pedagogical interventions or teaching purposes that are hard or impossible to know ahead of time. To address this, the best thing the ENVISAGE project can do is to maximally empower the teacher in the implementation of analytics, but supporting flexibility in use. The ENVISAGE project attempts to maximize flexibility in the implemented methods by taking a three-pronged approach to analytics:

1. Event tracking is designed to communicate the maximal amount of information about not only the choices that learning make in the virtual labs, but also the context for these choices, by including as much information about the labs when choices were made.

2. High-level metrics are defined with a strong anchoring in pedagogical theory to make them as generically useful, regardless of pedagogical interest.

3. Unsupervised, supervised, and generative machine learning is implemented to provide information within the context of each virtual lab, but leaves interpretation and intervention selection at the discretion of the authoring tool user, who is assumed to have the most relevant domain knowledge.

5 Chosen Machine Learning Algorithms

This section describes the machine learning algorithms that are being explored for predictive analytics and course adaptation methods in ENVISAGE. Informed by the review in the sections above, we are choosing to explore a number of player modeling approaches within unsupervised, supervised, and generative methods. Some of these are tailored to communicate back to the users of the authoring tool via visualizations also used for shallow analytics, while others are centered on providing information contextualized within the actual virtual labs which might later be combined with contextual visualization techniques such as heatmaps.

5.1 Unsupervised Modeling

One goal of the ENVISAGE project is to understand how different students' behaviors are indicative of different groupings within e.g. the whole student base or particular classes. In the terminology of Educational Data Mining, this is a Structure Discovery problem, which is a well-known class of problems. For both Game Analytics and Educational Data Analysis, this is typically addressed by applying clustering methods, that partition observations into groups.

Two clustering algorithms will be attempted for the data sets collected from the virtual labs, k-means and archetypal analysis. K-means allows for identifying groups based on typical behavior. Archetypal analysis, on the other hand, allows for identifying groups based on extreme behavior. We predict that both kinds of groupings may be of interest to teachers adapting virtual labs to suit their needs.

5.1.1 K-means Clustering

For finding groups, the k-means clustering algorithm is a good candidate, which will be explored for the virtual labs in the project. k-Means partitions data into groups, or clusters, based on the distance between observations, given a predefined distance function, and clusters them around representative points, known as centroids. These points may be calculated using either a mean measure or a median measure. It requires us to define a number of clusters for this grouping and can provide metrics of the appropriateness of different choices of numbers.

The algorithm starts initializing a number of random cluster centers equal to the number selected by the use, and then assigns each of the observation to the cluster which is closer (see Eq. 1):

$$S_i^{(t)} = \big\{ x_p : \big\| x_p - \mu_i^{(t)} \big\|^2 \le \big\| x_p - \mu_j^{(t)} \big\|^2 \; \forall j, 1 \le j \le k \big\} \tag{1}$$

It then proceeds to update the centers based on the memberships in these newly determined cluster groups (see Eq. 2).

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \tag{2}$$

.

The goal is to minimize the objective function (see 3) and the process of assigning observations to clusters and updating the centroids may continue until no improvement is seen in the objective function (or until no more computational resources are available):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2 \tag{3}$$

The k-means algorithm has been used extensively and successfully in game data mining to aggregate multi-dimensional data describing player behavior and grouping players based on these behaviors.

When combined with e.g Principal Component Analysis, it also lends itself well to visualization and allows for showing how different clusters are distributed relative to one another along a reduced number of abstract dimensions, as shown in e.g. Figure 1.

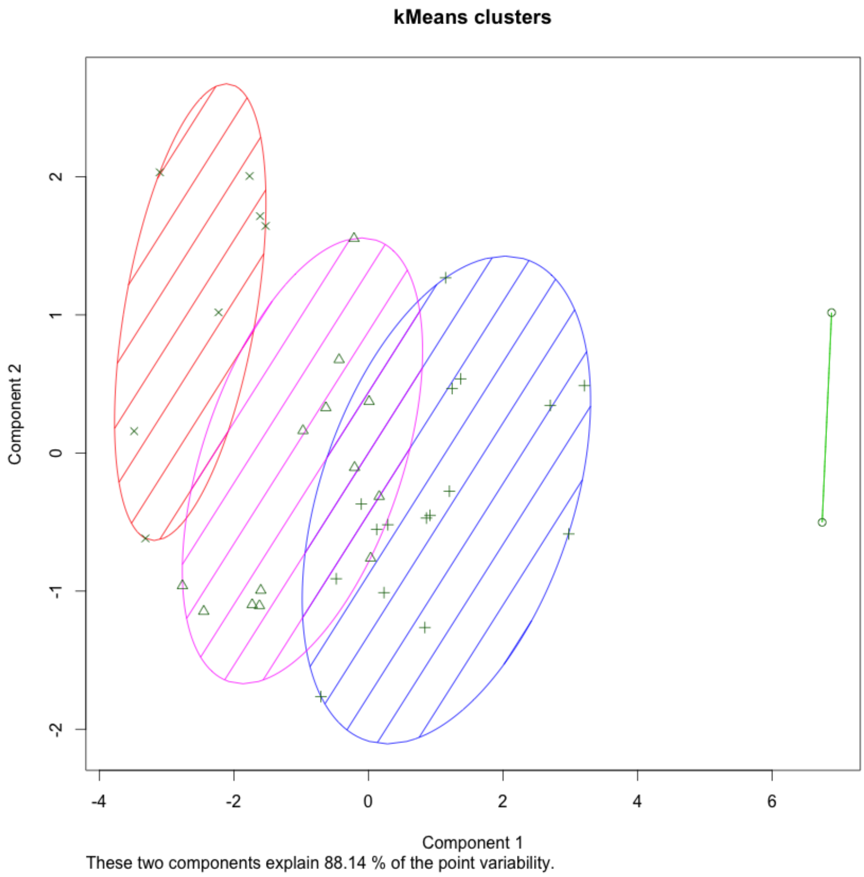

Figure 1: Example of k-means clustering of player behavior obtained from the game described in [9]. The figure delineates the player base into three groups, exhibiting different in-game behaviors.

5.1.2 Archetypal Clustering

Where k-means and the similar k-medoids focus on identifying groups around average behavior in the data, archetypal analysis is focused on identifying extreme examples in the data. The algorithm works by drawing the minimally possible convex hull around all the observed data points.

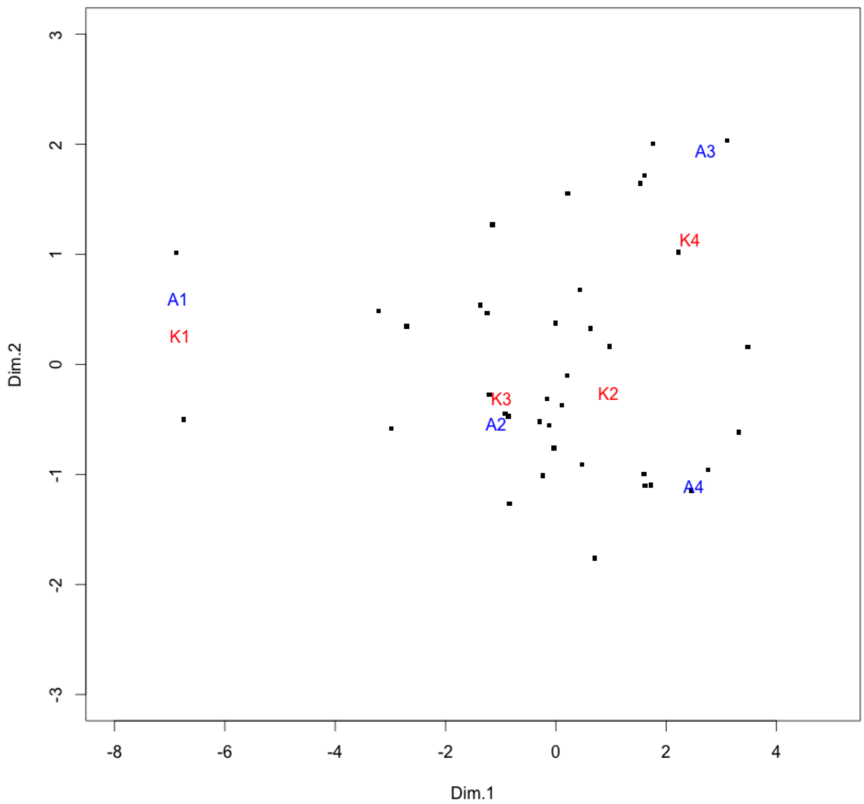

Figure 2: A comparison between cluster location on the same data set from a game when using either k-Means (indicated by K's) or archetypal analysis (indicated by A's).

Using this hull, the algorithm searches for linear combinations of the observed data points that minimize 4 to determine coefficients that allow the data to be represented by the archetypes [2].

$$\text{arg min } S, H\frac{1}{2}||X - XSH||_F^2 \tag{4}$$

Observations are then labeled according to their closeness to these archetypes, using a distance function, much akin to the way observations are labeled in k-Means.

When used in combination with k-means, archetypal analysis provides a useful alternative perspective that allows the user to see hypothetical extreme examples. This can help the user understand the overall directions of the behavior that the players of a game or the users of a digital learning environment are exhibiting.

Figure 2 shows a comparison of cluster centers found using k-Means and archetypal analysis, respectively, when applied to the same dataset of player actions in a game.

5.2 Supervised Modeling

Supervised modeling, in machine learning terms, corresponds to the Prediction Models category of Educational Data mining. The fundamental premise of the approach is to configure/train a model from an existing dataset composed of input/output variables, and using this trained model later to predict values for new observations as needed. In the instance of ENVISAGE, this could means predicting PISA-group membership based on performance in a virtual lab, or it could mean predicting subsequent test scores based off variables summarizing behavior in a virtual lab. A plethora of supervised modeling techniques exist, ranging from simple regression to highly complex deep neural network models. Choosing which one to implement may depend on the nature of the data in question and the particular modeling problem. For ENVISAGE, however, we are focusing on choosing a generally applicable method that can be expected to work well for most supervised modeling problems. For that reason we are initially applying supervised modeling through neural networks. Neural networks are characterized by being able to approximate any differentiable function, provided they are large enough and given enough training data and time [3], and thus may work as a general catch-all approach for most necessary supervised modeling in the ENVISAGE project. The disadvantage with using neural networks for supervised modeling is that the trained models are generally opaque to human inspection. I.e. other analytical methods must be used in order to understand the model represented in the network. This disadvantage has little impact on the ENVISAGE project, however, as only the outputs of the trained networks will be made available to the end users in the form of teachers.

5.3 Generative Modeling

Reinforcement learning will be used to attempt the learning of agent policies for simulating player behavior in the ENVISAGE virtual labs. Prior research has shown that this approach can be used to learn and generate user behavior shaped by preferences, using both evolutionary methods [9] as well as reinforcement learning [5]. The first approach that will be attempted will be a transfer of the method outlined in [9].

In this method, a configurable game-playing agent is developed and optimized to mimic human behavior through evolutionary computation. Agents may be evolved to match travel paths from single players or to match groups of players belonging to the same group, as obtained from e.g. the clustering algorithms outlined above.

In [9], this configurable agent is obtained by building the agent around a decision-making neural network that decides what action to take for each step of the simulation.

The agent is connected to a simulation of the game, or in this case, the digital learning environment. Simultaneously, action traces from actual human players/users are collected and organized in decision trees, representing the players('s) travel path through the digital environment. The weights of the neural-network are then optimized through evolutionary computation [13]. The agent is presented with all the contexts in which the human player made decisions in the travel path, and is asked for its corresponding decision. For each decision that matches the human decision the agent is awarded a point. Finally, when no more observations remain, the agent's *Action Agreement Ratio* (AAR) relative to the human player is calculated as shown in 5:

$$AAR = \frac{N_{match}}{N_{decision}} \tag{5}$$

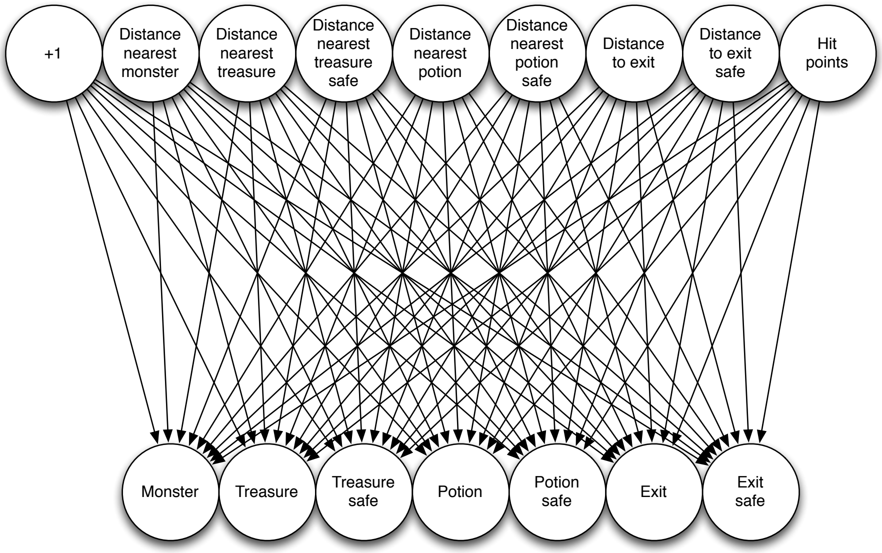

Figure 3: Example of an agent-controlling network, described in [9]. Similar networks will be experimentally attempted for several of the ENVISAGE games, particularly the Wind Energy Lab and the Chemistry Lab.

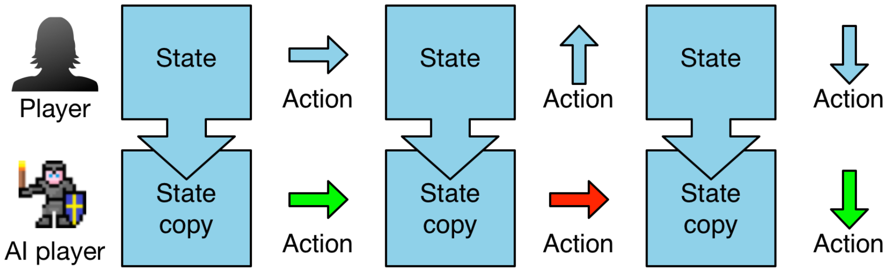

Figure 4: Illustration of how simulations and action comparisons can be used to calculate the AAR value between the actions of a player and the actions of a game-playing agent supposed to represent that player.

The action agreement ratio is then used as the fitness value for the agent in the evolutionary algorithm. This evolutionary process is the run to maximize the AAR value across all the travel paths that the agent is targeted to represent.

$$\arg\max AAR \tag{6}$$

The general principle is illustrated in Figure 4.

6 Software Architecture

The general software architecture delivering predictive analytics to the users of the authoring tools is described in D4.2 of the ENVISAGE project [17]. Here we describe the used machine learning libraries, and their purpose in the predictive analytics implementation.

6.1 Machine Learning Libraries and Implementation

The software stack shown in Table 1 is used to implement and explore the techniques described in this section:

Table 1: Software libraries used for predictive analytics and course adaptation methods.

Software	Purpose	Availability
pandas	Provides utilities for data acquisition and manipulation in preparation for modeling.	http://pandas.pydata.org/
scikit-learn	Provides the k-means and k-mediods algorithms.	http://scikit-learn.org/
rpy2 and R	Provides the archetypal analysis algorithm in R and provides this to the general implementation via rpy2.	https://rpy2.bitbucket.io/ and https://cran.r-project.org/
pyevolve	Provides utilities for evolutionary algorithms.	http://pyevolve.sourceforge.net/
keras and TensorFlow	Provides neural networks for supervised learning.	https://keras.io/ and https://www.tensorflow.org/

The implementations used to experimentally implement the unsupervised, supervised, and generative models described in this document can be found at the following URL:

`https://github.com/Envisage-H2020`

7 Conclusion

In this deliverable we briefly reviewed the state-of-the-art in game analytics and learning analytics as they relate to the ENVISAGE project. We identified how these approaches and discussions relate to the unique challenges faced by the ENVISAGE project, and how the identified principles can be applied in the context of learning analytics. Additionally, we identified the major technical steps involved in game analytics and related them to the ENVISAGE project in order to understand how they translate into Learning Analytics. Using insights into the problem of Learning Analytics gained from the other deliverables of ENVISAGE, we identified a number of specific technical approaches that can be employed in the ENVISAGE project to analyze, predict user behavior, and assist teachers in adapting the virtual labs of the project accordingly. The approaches were described under three general headings: unsupervised modeling, supervised modeling, and generative modeling. Whether all of these approaches will be technically or logistically feasible to implement under the ENVISAGE project remains to be seen, but all of them were identified as approaches potentially valuable to users of the ENVISAGE authoring tool. Finally, we described the software stack underpinning the modeling initiatives described here and how the deployment of existing open-source, scientific frameworks allows us to build a deep analytics platform for the ENVISAGE project in the most efficient way.

References

[1] Ryan Shaun Baker and Paul Salvador Inventado. Educational data mining and learning analytics. In *Learning analytics*, pages 61–75. Springer, 2014.

[2] Christian Bauckhage and Christian Thurau. Making archetypal analysis practical. In *DAGM-Symposium*, pages 272–281. Springer, 2009.

[3] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] Yun-Gyung Cheong, Rilla Khaled, Christoffer Holmgrd, and Georgios N Yannakakis. Serious games for teaching conflict resolution: Modeling conflict dynamics. In *Conflict and Multimodal Communication*, pages 449–475. Springer, 2015.

[5] Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*, 2017.

[6] Adele Cutler and Leo Breiman. Archetypal analysis. *Technometrics*, 36(4):338–347, 1994.

[7] Anders Drachen, Alessandro Canossa, and Georgios N Yannakakis. Player modeling using self-organization in tomb raider: Underworld. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 1–8. IEEE, 2009.

[8] Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. *Game Analytics*. Springer, 2016.

[9] Christoffer Holmgård, Antonios Liapis, Julian Togelius, and Georgios N Yannakakis. Evolving personas for player decision modeling. In *Computational Intelligence and Games (CIG), 2014 IEEE Conference on*, pages 1–8. IEEE, 2014.

[10] Christoffer Holmgård and Fabian Hadiji. D2.3 visualization strategies for course progress reports. In *ENVISAGE*. 2017.

[11] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[12] Georgios Mavromanolakis, Pavlos Koulouris, Sofoklis Sotiriou, Giannis Chantas, and Spiros Nikolopoulos. D1.2 data structure and functional requirements. In *ENVISAGE*. 2017.

[13] R Poli, WB Langdon, and NF McPhee. A field guide to genetic programming (with contributions by jr koza)(2008). *Published via http://lulu. com*, 2008.

[14] George Siemens and Ryan SJ d Baker. Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 252–254. ACM, 2012.

[15] Adam M Smith, Erik Andersen, Michael Mateas, and Zoran Popović. A case study of expressively constrainable level design automation tools for a puzzle game. In *Proceedings of the International Conference on the Foundations of Digital Games*, pages 156–163. ACM, 2012.

[16] Sofoklis Sotiriou, Pavlos Koulouris, and Georgios Mavromanolakis. D1.1 educational scenarios and stakeholder analysis. In *ENVISAGE*. 2017.

[17] Dimitrios Ververidis, Stathis Nikolaidis, Anastasios Papazoglou, Christoffer Holmgård Marc Mueller, and Fabian Hadiji. D4.2 - first version of the "virtual labs" authoring tool. In *ENVISAGE*. 2017.

[18] Georgios N Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. Player modeling. In *Dagstuhl Follow-Ups*, volume 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[19] Georgios N Yannakakis, Julian Togelius, Rilla Khaled, Arnav Jhala, Kostas Karpouzis, Ana Paiva, and Asimina Vasalou. Siren: Towards adaptive serious games for teaching conflict resolution. *Proceedings of ECGBL*, pages 412–417, 2010.