

## DATA SEGMENTATION USING CLUSTERING AND DECISION TREE

5

### CROSS-REFERENCES TO RELATED APPLICATION(S)

**[0001]** This application claims priority to US Provisional Application No. 63/480,364 filed January 18, 2023, which is incorporated by reference herein in its entirety.

### BACKGROUND

10 **[0002]** Machine learning (ML) is an area of artificial intelligence (AI) where computers have the capability to learn without being explicitly programmed. There are different types of ML techniques including supervised learning techniques, unsupervised learning techniques, and others. In a supervised learning technique, an ML model is trained using training data, where the training data includes multiple training examples, each training example including an input  
15 and a known output corresponding to the input. In an unsupervised learning technique, an ML model or algorithm is provided with unlabeled data, and is tasked to analyze and find patterns in the unlabeled data. The examples of unsupervised learning technique are dimension reduction and clustering.

**[0003]** Another approach is using unsupervised ML clustering where data points with similar  
20 features are assigned into clusters. For example, in the clustering, the ML model is tasked to segment the data points into a number of groups so that data points in the same groups are more similar to other data points in the same group than those in other groups. The ML clustering has several advantages over the rule-based approach, e.g., the segmentation may be performed in multiple dimensions, and variances within each resulting cluster are very small.

25 **[0004]** However, in some domains (e.g., medical, financial, security), the ML model for segmentation is trained on the confidential data. It is difficult to transmit the ML model for segmentation that contains confidential data because of data security constraints.

### SUMMARY

30 **[0005]** Embodiments of the present disclosure provide systems, methods, and apparatuses for addressing the above problems by performing unsupervised ML clustering to segment the data into clusters, and then training a rule-based decision tree using the clusters from the

clustering, so that the rules of the decision tree, when applied on the same or similar data, replicate the segmentation as if performed by the unsupervised ML clustering.

5 [0006] Some embodiments of the present disclosure include a method performed by one or more processors of a computer system. Sets of historical access requests for resources are received from a plurality of requestors, where each historical access request of each of the sets of historical access requests is associated with a same requestor having a requestor identifier, each requestor identifier being associated with requestor features. Requestor identifiers of the plurality of requestors are clustered to obtain a plurality of clusters for the requestor identifiers, the clustering using a plurality of requestor features associated with requestor identifiers of the 10 plurality of requestors. A decision tree that segments the requestor identifiers into segments is trained using the plurality of clusters so that the segments respectively match the plurality of clusters obtained from the clustering, where the decision tree includes nodes, each of the nodes storing a rule among a plurality of rules, where an application of the decision tree on the plurality of requestor features causes a segmentation being performed, the segmentation 15 replicating an application of the clustering on the plurality of requestor features to obtain the plurality of clusters. The plurality of rules and an architecture of the decision tree are transmitted to another computer system, thereby allowing the other computer system to perform segmentation using access requests for resources from requestors, where the access requests are stored locally on the other computer system.

20 [0007] These and other embodiments of the disclosure are described in detail below. For example, other embodiments are directed to systems, devices, and computer-readable media associated with methods described herein.

[0008] A better understanding of the nature and advantages of embodiments of the present disclosure may be gained with reference to the following detailed description and the 25 accompanying drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The patent or application file contains at least one drawing executed in color. Copies of this patent or patent application publication with color drawing(s) will be provided by the Office upon request and payment of the necessary fee.

30 [0010] FIG. 1 shows a resource security system for authorizing access to resources, in accordance with some embodiments of the present invention.

[0011] FIG. 2 shows a block diagram of an exemplary data segmentation system, in accordance with some embodiments of the present invention.

[0012] FIG. 3 shows an example of K-means clustering.

[0013] FIG. 4A shows an example of Euclidean distance.

[0014] FIG. 4B shows an example of K-means clustering using centroids.

[0015] FIG. 5 shows an example of K-means clustering, in accordance with some embodiments of the present invention.

5 [0016] FIG. 6 shows an example of a decision tree, according to embodiments of the present invention.

[0017] FIG. 7 shows an example of a decision tree, according to embodiments of the present invention.

10 [0018] FIG. 8 shows an exemplary flowchart of a method, according to embodiments of the present invention.

[0019] FIG. 9 shows an exemplary flowchart of a method, according to embodiments of the present invention.

[0020] FIG. 10 shows a block diagram of an exemplary computer apparatus, in accordance with some embodiments of the present invention.

15 TERMS

[0021] Prior to discussing embodiments of the disclosure, description of some terms may be helpful in understanding embodiments of the disclosure.

20 [0022] The term “resource” generally refers to any asset that may be used or consumed. For example, the resource may be an electronic resource (e.g., stored data, received data, a computer account, a network-based account, an email inbox), a physical resource (e.g., a tangible object, a building, a safe, or a physical location), or other electronic communications between computers (e.g., a communication signal corresponding to an account for performing a transaction). Thus, a physical resource can be a physical object.

25 [0023] The term “resource provider” may refer to an entity that can provide resources such as goods, services, information, and/or access. Examples of a resource provider includes merchants, access devices, secure data access points, etc. A “merchant” may typically be an entity that engages in transactions and can sell goods or services, or provide access to goods or services.

30 [0024] The term “access request” (also referred to as an “authentication request”) generally refers to a request to access a resource. The access request may be received from a requesting computer, a user device, or a resource computer, for example. The access request may include authentication information (also referred to as authorization information), such as a username, resource identifier (ID), or password. The access request may also include and access request

parameters, such as an access request identifier, a resource identifier, a timestamp, a date, a device or computer identifier, a geo-location, or any other suitable information. A real-time access request occurs when access to a resource is desired at the time the request is made. In such situations, it is desirable to provide a quick determination for whether to provide access  
5 to the resource.

**[0025]** The term “access rule” may include any procedure or definition used to determine an access rule outcome for an access request based on certain criteria. In some embodiments, the rule may include one or more rule conditions and an associated rule outcome. A “rule condition” may specify a logical expression describing the circumstances under which the  
10 outcome is determined for the rule. A condition of the access rule may involve authentication information, as well as request parameters. For example, the authentication information can be required to sufficiently correspond to information categorized as legitimate, e.g., based on a match to critical nodes of a data structure and/or to a sufficient number of nodes. A condition can require a specific parameter value, a parameter value to be within a certain range, a  
15 parameter value being above or below a threshold, or any combination thereof.

**[0026]** The term “server computer” may include a powerful computer or cluster of computers. For example, the server computer can be a large mainframe, a minicomputer cluster, or a group of computers functioning as a unit. In one example, the server computer may be a database server coupled to a web server. The server computer may be coupled to a database  
20 and may include any hardware, software, other logic, or combination of the preceding for servicing the requests from one or more other computers.

**[0027]** The term “computer system” may generally refer to a system including one or more server computers coupled to one or more databases.

**[0028]** The term “memory” may refer to any suitable device or devices that may store  
25 electronic data. A suitable memory may include a non-transitory computer-readable medium that stores instructions that can be executed by a processor to implement a desired method. Examples of memories include one or more memory chips, disk drives, etc. Such memories may operate using any suitable electrical, optical, and/or magnetic mode of operation.

**[0029]** The term “processor” may refer to any suitable data computation device or devices.  
30 A processor may include one or more microprocessors working together to accomplish a desired function. The processor may include a CPU that includes at least one high-speed data processor adequate to execute program components for executing user and/or system generated requests. The CPU may be a microprocessor such as AMD’s Athlon, Duron and/or Opteron;

IBM and/or Motorola's PowerPC; IBM's and Sony's Cell processor; Intel's Celeron, Itanium, Pentium, Xenon, and/or XScale; and/or the like processor(s).

**[0030]** The term "machine learning model" may refer to a program, file, method, or process, used to perform some function on data, based on knowledge "learned" during a training phase.

5 For example, a machine learning model can be used to classify feature vectors as normal or anomalous. In "supervised learning," during a training phase, a machine learning model can learn correlations between features contained in feature vectors and associated labels. After training, the machine learning model can receive unlabeled feature vectors and generate the corresponding labels. For example, during training, a machine learning model can evaluate  
10 labeled images of dogs, then after training, the machine learning model can evaluate unlabeled images, in order to determine if those images are of dogs. In "unsupervised learning," an ML model or algorithm is provided with unlabeled data, and is tasked to analyze and find patterns in the unlabeled data.

**[0031]** Machine learning models may be defined by "parameter sets," including  
15 "parameters," which may refer to numerical or other measurable factors that define a system (e.g., the machine learning model) or the condition of its operation. In some cases, training a machine learning model may include identifying the parameter set that results in the best performance by the machine learning model. This can be accomplished using a "loss function," which may refer to a function that relates a model parameter set to a "loss value" or "error  
20 value," a metric that relates the performance of a machine learning model to its expected or desired performance.

**[0032]** The term "entities" may include things with distinct and independent existence. For example, entities may include people, organizations (e.g., partnerships and businesses), computers, and computer networks, among others. An entity can communicate or interact with  
25 its environment. Further, an entity can operate, interface, or interact with a computer or computer network during the course of its existence.

**[0033]** The term "issuer" refers to a business entity (e.g., a bank) that maintains an account for a user. An issuer may also issue payment credentials stored on a user device, such as a cellular telephone, smart card, tablet, or laptop to the consumer.

30 **[0034]** The term "payment processing network" may include data processing subsystems, networks, and operations used to support and deliver authorization services, exception file services, and clearing and settlement services. The payment processing network may use any suitable wired or wireless network, including the Internet.

[0035] The term “acquirer” refers to a business entity (e.g., a commercial bank) that has a business relationship with a particular merchant or other entity. Some entities can perform both issuer and acquirer functions. Some embodiments may encompass such single entity issuer-acquirers.

- 5 [0036] The term “authorizing entity” refers to an entity that authorizes an access request. Examples of an authorizing entity may be an issuer, a governmental agency, a document repository, an access administrator, etc.

- [0037] Clustering refers to a data mining technique which groups unlabeled data based on their similarities or differences. Clustering algorithms are used to process raw, unclassified data  
10 objects into groups represented by structures or patterns in the information. Grouping similar entities together help profile the attributes of different groups. In other words, it provides insight into underlying patterns of different groups. There are many applications of grouping unlabeled data, for example, identification of different groups/segments of customers and marketing each group in a different way to maximize the revenue. Another example is grouping documents  
15 together that belong to the similar topics etc.

- [0038] K-means clustering is an example of a clustering method where data points are assigned into K groups, where K represents the number of clusters based on the distance from each group’s centroid. The data points closest to a given centroid will be clustered under the same category. A larger K value will be indicative of smaller groupings with more granularity  
20 whereas a smaller K value will have larger groupings and less granularity. K-means clustering may be used in market segmentation, document clustering, image segmentation, and image compression.

- [0039] Hierarchical clustering, also known as hierarchical cluster analysis (HCA), is an unsupervised clustering algorithm that can be agglomerative or divisive. Agglomerative  
25 clustering is considered a “bottoms-up approach.” Its data points are isolated as separate groupings initially, and then they are merged together iteratively on the basis of similarity until one cluster has been achieved.

- [0040] A probabilistic model is an unsupervised technique that helps to solve density estimation or “soft” clustering problems. In probabilistic clustering, data points are clustered  
30 based on the likelihood that they belong to a particular distribution. The Gaussian Mixture Model (GMM) is the one of the most commonly used probabilistic clustering methods.

## DETAILED DESCRIPTION

[0041] Embodiments can provide systems, methods, and apparatuses for segmenting data. More particularly, embodiments are for performing unsupervised ML clustering to segment the access requests into clusters, and then training a rule-based decision tree using the clusters from the clustering, so that the rules of the decision tree, when applied on the access requests, replicate the segmentation as if performed by the unsupervised ML clustering.

[0042] One approach for segmentation is the threshold or rule-based approach, where a user selects *a priori* thresholds and divides the data points accordingly. However, this approach leads to very large variances among the data points found in each segment. Further, it is difficult to perform the segmentation in more than two dimensions.

[0043] Another approach is using unsupervised ML clustering where data points with similar features are assigned into clusters. For example, in the clustering, the ML model is tasked to segment the data points into a number of groups so that data points in the same groups are more similar to other data points in the same group than those in other groups. The ML clustering has several advantages over the rule-based approach, e.g., the segmentation may be performed in multiple dimensions, and variances within each resulting cluster are very small.

[0044] However, in some domains (e.g., medical, financial, security), the ML model for segmentation is trained on the confidential data. Transmitting the trained ML model for segmentation that contains confidential data is prohibitive for many customers.

[0045] In certain embodiments, sets of historical access requests that are associated with requestors are used to collect information about requestor features of the requestors, where each historical access request of each of the sets of historical access requests is associated with a same requestor having a requestor identifier. Requestor identifiers are clustered based on selected features using unsupervised ML clustering. As a result of clustering, clusters for the requestor identifiers may be obtained in multiple dimensions, where the variances within each cluster are very small as compared to the existing rule-based segmentation.

[0046] Then, training is performed on a decision tree using the clusters, to derive rules and thresholds from the data of the clusters, so that the trained decision tree can segment the requestor identifiers into segments that respectively match the clusters obtained from the clustering. The rules and the architecture of the decision tree can be transmitted to a customer side computer system, without transmitting the requestor identifiers, e.g., primary account numbers (PANs) of users. The decision tree can be applied to segment the locally stored access

requests by using the requestor features associated with the requestor identifiers internal to the customer computer system. An application of the decision tree causes a segmentation being performed that replicates an application of the clustering on the requestor identifiers into the clusters.

5    **[0047]** As such, the segmentation, which replicates the clustering and possesses the advantages of the clustering, may be performed locally by using the decision tree trained on the results of prior clustering, without the need to transmit the model for segmentation and associated PANs. Thus, the novel techniques preserve the users privacy while using lower computational resources as compared to where the complex model containing confidential  
10   information is transmitted and/or used.

**[0048]** Further, the segmentation using the decision tree structure according to the novel techniques may be done efficiently and accurately in multiple dimensions as compared to the rule-based segmentation where the segmentation is performed only in a limited number of dimensions by using *a priori* thresholds and produces segments with large variances.

15   **[0049]** Authentication using a resource security system is first discussed, followed by a description of the data categorization according to embodiments.

## **I.     ACCESSING A PROTECTED RESOURCE**

**[0050]** Generally, access requests for a computer resource or account (e.g., transactions over the Internet) go through an authentication system to determine whether the transaction is  
20   authorized or rejected, for example, due to being fraudulent. Thus, a resource security system may receive requests to access a resource. The resource security system may include an access server for determining an outcome for the access request based on access rules. An exemplary resource security system is described in further detail below.

**[0051]** FIG. 1 shows a resource security system 100 for authorizing access to resources, in  
25   accordance with some embodiments. The resource security system 100 may be used to provide authorized users (e.g., via authentication) access to a resource while denying access to unauthorized users. In addition, the resource security system 100 may be used to deny fraudulent access requests that appear to be legitimate access requests of authorized users. The resource security system 100 may implement access rules to identify fraudulent access requests  
30   based on parameters of the access request. Such parameter may correspond to fields (nodes) of a data structure that is used to distinguish fraudulent access requests from authentic access requests.

[0052] The resource security system 100 includes a resource computer 110. The resource computer 110 may control access to a physical resource 118, such as a building or a lockbox, or an electronic resource 116, such as a local computer account, digital files or documents, a network database, an email inbox, a payment account, or a website login. In some  
5 embodiments, the resource computer may be a webserver, an email server, or a server of an account issuer. The resource computer 110 may receive an access request from a user 140 via a user device 150 (e.g., a computer or a mobile phone) of the user 140. The resource computer 110 may also receive the access request from the user 140 via a request computer 170 coupled with an access device 160 (e.g., a keypad or a terminal). In some embodiments, the request  
10 computer 170 may be a resource provider. For example, the request computer 170 and the resource computer 110 may be the same, where the access request from the user 140 is generated directly at the resource computer 110.

[0053] The access device 160 and the user device 150 may include a user input interface such as a keypad, a keyboard, a fingerprint reader, a retina scanner, any other type of biometric  
15 reader, a magnetic stripe reader, a chip card reader, a radio frequency identification reader, or a wireless or contactless communication interface, for example. The user 140 may input authentication information into the access device 160 or the user device 150 to access the resource. Authentication information may also be provided by the access device 160 and/or the user device 150. The authentication information may include, for example, one or more  
20 data elements of a username, an account number, a token, a password, a personal identification number, a signature, a digital certificate, an email address, a phone number, a physical address, and a network address. The data elements may be labeled as corresponding to a particular field, e.g., that a particular data element is an email address. In response to receiving authentication information input by the user 140, the user device 150 or the request computer  
25 170 may send an access request, including authentication information, to the resource computer 110 along with one or more parameters of the access request.

[0054] In one example, the user 140 may enter one or more of an account number, a personal identification number, and password into the access device 160, to request access to a physical resource (e.g., to open a locked security door in order to access a building or a lockbox) and  
30 the request computer 170 may generate and send an access request to the resource computer 110 to request access to the resource. In another example, the user 140 may operate the user device 150 to request that the resource computer 110 provide access to the electronic resource 116 (e.g., a website or a file) that is hosted by the resource computer 110. In another example, the user device 150 may send an access request (e.g., an email) to the resource computer 110

(e.g., an email server) in order to provide data to the electronic resource 116 (e.g., deliver the email to an inbox). In another example, the user 140 may provide an account number and/or a personal identification number to an access device 160 in order to request access to a resource (e.g., a payment account) for conducting a transaction.

5   **[0055]** In some embodiments, the resource computer 110 may verify the authentication information of the access request based on information stored at the request computer 170. In other embodiments, the request computer 170 may verify the authentication information of the access request based on information stored at the resource computer 110.

10   **[0056]** The resource computer 110 may receive the request substantially in real-time (e.g., account for delays computer processing and electronic communication). Once the access request is received, the resource computer 110 may determine parameters of the access request. In some embodiments, the parameters may be provided by the user device 150 or the request computer 170. For example, the parameters may include one or more of: a time that the access request was received, a day of the week that the access request was received, the source-  
15   location of the access request, the amount of resources requested, an identifier of the resource being request, an identifier of the user 140, the access device 160, the user device 150, the request computer 170, a location of the user 140, the access device 160, the user device 150, the request computer 170, an indication of when, where, or how the access request is received by the resource computer 110, an indication of when, where, or how the access request is sent  
20   by the user 140 or the user device 150, an indication of the requested use of the electronic resource 116 or the physical resource 118, and an indication of the type, status, amount, or form of the resource being requested. In other embodiments, the request computer 170 or the access server 120 may determine the parameters of the access request.

25   **[0057]** The resource computer 110 or the request computer 170 may send the parameters of the access request to the access server 120 in order to determine whether the access request is fraudulent. The access server 120 may store one or more access rules 122 for identifying an illegal access request. Each of the access rules 122 may include one or more conditions corresponding to one or more parameters of the access request. The access server 120 may determine an access request outcome indicating whether the access request should be accepted  
30   (e.g., access to the resource granted), rejected (e.g., access to the resource denied), or reviewed by comparing the access rules 122 to the parameters of the access request as further described below. In some embodiments, instead of determining an access request outcome, the access server 120 may determine an evaluation score based on outcomes of the access rules. The evaluation score may indicate the risk or likelihood of the access require being fraudulent. If

the evaluation score indicates that the access request is likely to be fraudulent, then the access server 120 may reject the access request.

**[0058]** The access server 120 may send the indication of the access request outcome to the resource computer 110 (e.g., accept, reject, review, accept and review, or reject and review).

5 In some embodiments, the access server 120 may send the evaluation score to the resource computer 110 instead. The resource computer 110 may then grant or deny access to the resource based on the indication of the access request outcome or based on the evaluation score. The resource computer 110 may also initiate a review process for the access request.

**[0059]** In some embodiments, the access server 120 may be remotely accessed by an administrator for configuration. The access server 120 may store data in a secure environment and implement user privileges and user role management for accessing different types of stored data. For example, user privileges may be set to enable users to perform one or more of the following operations: view logs of received access request, view logs of access request outcomes, enable or disable the execution of the access rules 122, update or modify the access rules 122, change certain access request outcomes. Different privileges may be set for different users.

**[0060]** The resource computer 110 may store access request information for each access requests that it receives. The access request information may include authentication information and/or the parameters of each of the access requests. The access request information may also include an indication of the access request outcome for the access request, e.g., whether access request was actually fraudulent or not. The resource computer 110 may also store validity information corresponding to each access request. The validity information for an access request may be initially based on its access request outcome. The validity information may be updated based on whether the access request is reported to be fraudulent.

25 In some embodiments, the access server 120 or the request computer 170 may store the access request information and the validity information.

## II. DATA SEGMENTATION SYSTEM

**[0061]** As described in detail below, embodiments of the disclosure can use historical access requests for resources from the requestors, to cluster requestor identifiers of the plurality of requestors using the requestor features associated with the requestor identifiers, and train a decision tree using data points of the clusters. As a result of the training, the decision tree can be generated that segments the requestor identifiers into segments that respectively match the clusters obtained from the clustering, so that the data associated with the decision tree can be

transmitted to the customer, where the customer can replicate the clustering by using the decision tree on the locally stored access requests.

**[0062]** In embodiments, data points of the clusters correspond to the requestor identifiers. Thus, for convenience of description of embodiments, the data points of the clusters are referred to as the requestor identifiers.

**[0063]** FIG. 2 is a simplified block diagram of a data segmentation system 200 according to certain embodiments. The data segmentation system 200 may be implemented using one or more computer systems, each computer system having one or more processors. The data segmentation system 200 may include multiple components and subsystems communicatively coupled to each other via one or more communication mechanisms. For example, in the embodiment shown in FIG. 2, the data segmentation system 200 includes a data sorting subsystem 202, a clustering subsystem 204, and a decision tree learning subsystem 206. These subsystems may be implemented as one or more computer systems. The systems, subsystems, and other components shown in FIG. 2 may be implemented in software (e.g., code, instructions, program) executed by one or more processing units (e.g., processors, cores) of the respective systems, using hardware, or combinations thereof. The software may be stored on a non-transitory storage medium (e.g., on a memory device). The data segmentation system 200 shown in FIG. 2 is merely an example and is not intended to unduly limit the scope of embodiments. Many variations, alternatives, and modifications are possible. For example, in some implementations, data segmentation system 200 may have more or fewer subsystems or components than those shown in FIG. 2, may combine two or more subsystems, or may have a different configuration or arrangement of subsystems. The data segmentation system 200 and subsystems shown in FIG. 2 may be implemented using one or more computer systems, such as the computer system shown in FIG. 10.

**[0064]** As shown in FIG. 2, the data segmentation system 200 also includes a storage subsystem 220 that may store the various data constructs and programs used by the data segmentation system 200. For example, the storage subsystem 220 may store historical access requests 222. However, this is not intended to be limiting. In alternative implementations, the historical access requests 222 may be stored in other memory storage locations (e.g., different databases) that are accessible to the data segmentation system 200, where these memory storage locations can be local to or remote from the data segmentation system 200. As

described in detail below, the data segmentation system 200 clusters the requestor identifiers associated with

the historical access requests 222 into clusters that are then used to arrive at the rules of the decision tree.

- 5   **[0065]** For example, the historical access requests 222 may be historical transaction data collected by a customer over time and made available to the storage subsystem 220 and/or the data segmentation system 200. The historical access requests 222 correspond to a plurality of users, e.g., resource requestors or requestors, and include historical transaction data corresponding to a variety of access requests for a variety of the resource categories. As an  
10 example, the historical access requests 222 may include sets of historical access requests corresponding to requestor accounts collected over a time period.

#### *A. Data Preparation*

- 15   **[0066]** With continuing reference to FIG. 2, the data sorting subsystem 202 receives the historical access requests 222, e.g., transaction records for resources that correspond to the requestors. In certain implementations, the historical access requests 222 are arranged in sets including a first requestor dataset 224 to an Nth requestor dataset 226, where each historical access requests in each of the first requestor dataset 224 to the Nth requestor dataset 226 are  
20 respectively associated with a same requestor having a requestor identifier associated with requestor features or a requestor profile. E.g., each set of historical access requests may be associated with a same requestor identifier associated with the requestor features of one requestor.

- 25   **[0067]** In some embodiments, the data sorting subsystem 202 preprocesses the historical access requests 222 to exclude from further processing certain requestor identifiers. For example, the data sorting subsystem 202 can delete the historical access requests associated with the requestor identifiers for whom little activity is detected, e.g., for whom a very small number of monthly historical access requests is present, e.g., less than 1, the historical access requests associated with the requestor identifiers of requestors for whom the activity is detected  
30 for a shorter most recent time period than a predetermined time period, e.g., less than for three last months, and/or the historical access requests associated with the requestor identifiers for whom a great activity is detected, e.g., more than 99% of the total number of the historical access requests or average monthly number of the historical access requests.

[0068] The data sorting subsystem 202 then may divide the remaining requestor identifiers into a plurality of main groups based on one or more main features, for example, using an ML model or an ML algorithm that is capable of performing a segmentation.

5 [0069] In certain implementations, the data sorting subsystem 202 divides the requestor identifiers into two main groups based on one feature. In an example, the data sorting subsystem 202 divides the requestor identifiers into first requestor identifiers and second requestor identifiers based on a number of merchant categories associated with the historical requests corresponding to each requestor identifier.

10 [0070] In detail, each of the first requestor dataset 224 to the Nth requestor dataset 226 is associated with resource categories, e.g., merchant categories of the transaction records corresponding to the requestor identifiers respectively associated with each of the first requestor dataset 224 to the Nth requestor dataset 226. As an example, the resource categories may include clothing, groceries, travel, etc. The data sorting subsystem 202 forms a first requestor group 230, to include the first requestor identifiers, where each of first requestor  
15 identifiers is associated with a number of resource categories that is greater than or equal to a first threshold number, and a second requestor group 232, to include second requestor identifiers, where each of the second requestor identifiers is associated with a number of resource categories that is smaller than the first threshold number. However, this is not intended to be limiting. In some embodiments, the data sorting subsystem 202 forms the first requestor  
20 group 230 and the second requestor group 232 based on amount of money spent across the resource categories.

[0071] In certain implementations, where the data sorting subsystem 202 performs preprocessing, the data sorting subsystem 202 forms the first requestor group 230 including the first requestor identifiers that are approximately 97% of all first requestor identifiers, and the  
25 second requestor group 232 including the second requestor identifiers that are approximately 93% of all second requestor identifiers.

[0072] The data sorting subsystem 202 may store the first requestor group 230 and the second requestor group 232 in the storage subsystem 220.

30 [0073] However, the described above is not intended to be limiting. In some implementations, the data sorting subsystem 202 may be omitted and the first requestor identifiers associated with the first requestor group 230 and the second requestor identifiers

associated with the second requestor group 232 may be received by the data segmentation system 200 from an external device.

5 [0074] As described herein, the requestor identifiers are divided into two main groups based on one main feature; however, this is not intended to be limiting. In some embodiments, the requestor identifiers may be divided into a different number of main groups based on a plurality of main features.

[0075] As described in detail below, the features of the requestor identifiers associated with each of the main groups are used by the data segmentation system 200 to perform unsupervised clustering.

10

### *B. Clustering*

[0076] As described above, the data segmentation system 200 clusters the requestor identifiers associated with the historical access requests 222 into clusters that are then used to arrive at the rules of the decision tree.

15 [0077] With continuing reference to FIG. 2, the clustering subsystem 204 performs an unsupervised clustering, e.g., an unsupervised machine learning. Clustering involves grouping a set of objects into classes of similar objects. In some embodiments, the clustering subsystem 204 may perform K-means clustering.

#### **1. K-means Clustering**

20 [0078] FIG. 3 depicts an example of K-means clustering 300. In an example, data points 302, 304, and 306 may be clustered or grouped via K-means clustering 300 into clusters. In particular, the data points 302, 304, and 306 belong to clusters 308, 310, and 312, respectively.

[0079] In various embodiments, the data points correspond to the requestor identifiers, as described above. In an example, the data points may be represented as points or vectors in a multi-dimensional space. The locations of a respective data point in the multi-dimensional space may represent the respective point's data values, for example, amounts or certain characteristics. The data points' locations may quantify characteristics of the data. For example, in some embodiments, the closer the location values of respective data points, the more similar the data points may be.

30 [0080] The unsupervised learning process, e.g., K-means clustering 300, can operate based on a distance, such as a Euclidean distance, between the data points and their respective

clusters. The distance may be a measure of similarity, e.g., the smaller the distance between two data points, the more similar the data points may be. In some embodiments, the data segmentation system 200 may use other measures of similarity.

**[0081]** The unsupervised learning process, e.g., K-means clustering 300, can optimize, or locally optimize, the clusters such that the data points best match their respective clusters. In some embodiments, the number  $k$  of clusters may be automatically determined or user-specified in advance. In some embodiments, the data segmentation system 200 can optimize the number of clusters, the centroid locations of the clusters, and/or the composition of data points within each cluster, in order to achieve optimal clustering and learning.

## 2. Euclidean distance

**[0082]** In certain implementations, the distance described above may be a Euclidean distance.

**[0083]** FIG. 4A depicts an example of a Euclidean distance 400. In an example, the Euclidean distance 400 corresponds to the distance between a first point 402, e.g., a data point to be clustered, and a second point 404. In embodiments, the second point 404 may be a cluster centroid, e.g., a mean or median position of the coordinates of data points already associated with a particular cluster, or some other measure of the cluster's center.

**[0084]** The Euclidean distance 400 may be obtained by a  $d$ -dimensional distance formula, e.g., such as in standard Euclidean geometry:  $D_{Euc}(p, c) = \sum_{i=1}^d (p_i - c_i)^2$ , where  $D_{Euc}$  is the square of the Euclidean distance, the data points  $p \in \mathbb{Z}_{2^\ell}^d$ , and clusters  $c \in \mathbb{Z}_{2^\ell}^d$  are  $d$ -dimensional vectors with elements in  $\mathbb{Z}_{2^\ell}$ . In this formula,  $p$  and  $c$  are fixed to a particular data point and cluster, respectively, and the index  $i$  indexes a coordinate or component of the  $d$ -dimensional vector space.

**[0085]** In some embodiments, the distance may be another distance function or metric. For example, some other function of the coordinates of first point 402 and second point 404 may be used. For instance, different dimensions may be weighted differently for determining the distance.

**[0086]** FIG. 4B depicts an example of K-means clustering 300 based on a distance, e.g., a Euclidean distance, according to embodiments. In an example, the number  $K$  of clusters is 3, and, thus, three cluster centers 458, 460, and 462 are placed in the space.

[0087] In some embodiments, the data segmentation system 200 computes the Euclidean distance from all the data points, e.g., all of the data points 452, all of the data points 453, and all of the data points 456, to each cluster center, e.g., the cluster centers 458, 460, and 462. Taking as an example a data point 454, the data segmentation system 200 can compute the distance from a data point 454 to three clusters with the cluster centers 458, 460, and 462. Because the data point 454 has a shorter Euclidean distance to the cluster center 458 than to the cluster centers 460 and 462, the data point 454 may be assigned to the cluster with the cluster center 458. In the same manner, the distance from each data point to the cluster centers 458, 460, and 462 is calculated, and each data point is assigned to the closest cluster.

10 [0088] Next, the location of the cluster centers 458, 460, and 462 is recalculated as a mean of the data points assigned to it. For example, if the cluster centers 458, 460, 462 are centroids, the centroid locations can be recomputed.

[0089] In certain implementations, the process of the assigning the data points to the clusters and recomputing locations of the cluster centers is repeated until no more changes occur.

15 [0090] With reference again to FIG. 2, as described above, the clustering subsystem 204 performs an unsupervised clustering, to cluster the requestor identifiers of the requestors to obtain a plurality of clusters. The clustering subsystem 204 performs clustering using the requestor features associated with the requestor identifiers of the requestors. The requestor features may include, for each requestor identifier, at least one from among a first feature, a second feature, a third feature, a fourth feature, and a fifth feature. In an example, the first feature may correspond to a number of average monthly historical access requests per requestor identifier, the second feature may correspond to a percentage of benefit access requests out of a total number of historical access requests per requestor identifier, the third feature may correspond to a percentage of portfolio access requests out of a total number of historical access requests per requestor identifier, the fourth feature may correspond to a percentage of non-benefit access requests out of a total number of historical access requests per requestor identifier, and a fifth feature may correspond to a percentage of digitally-engaged access requests out of a total number of historical access requests per requestor identifier.

25 [0091] As an example, the benefit access requests are a category where the requestors may expect a benefit, e.g., cashback, provided by one of the parties to the transaction, e.g., an issuer, an acquirer, and/or a resource provider. The portfolio access requests are for resources provided by an issuer. E.g., if the issuer is United Airlines and a customer makes a United Airlines based

purchase, e.g., United Airlines airfare, then the access request of the customer may be classified as a portfolio access request. The digitally-engaged access requests are a category where the requestors conduct the transactions over the Internet, e.g., not face to face over the counter. These transactions may also be referred to as card-not-present (CNP).

5 **[0092]** As described above, in some implementations, the data sorting subsystem 202 may divide the requestor identifiers into the first requestor identifiers and the second requestor identifiers, where the first requestor group 230 and the second requestor group 232 are formed.

**[0093]** The clustering subsystem 204 may include a first clusterer 240 (e.g., a first clustering code) and a second clusterer 242 (e.g., a second clustering code). The first clusterer 240 may  
10 receive, as an input, the first requestor identifiers of the first requestor group 230 and the first requestor features associated with the first requestor identifiers, and cluster the first requestor identifiers of the first requestor group 230 into first clusters, using the first requestor features.

**[0094]** The second clusterer 242 may receive, as an input, the second requestor identifiers of the second requestor group 232 and the second requestor features associated with the second  
15 requestor identifiers, and cluster the second requestor identifiers into second clusters using the second requestor features associated with the second requestors. The first requestor features and the second requestor features may be the same, partially different, or completely different from each other.

**[0095]** In an embodiment, the first clusterer 240 and the second clusterer 242 may use K-  
20 means clustering. However, this is not intended to be limiting, and other methods of clustering known to those skilled in the relevant art may be used, e.g., hierarchical, probabilistic, etc.

**[0096]** In some embodiments, the clustering subsystem 204 may select a number K of the clusters by performing a Silhouette analysis. The Silhouette coefficient or Silhouette score is a measure of how similar a data point is within cluster (cohesion) compared to other clusters  
25 (separation). In Silhouette analysis, a range of K numbers is initially selected, e.g., 1 to 20. The Silhouette coefficient for a particular data point can be calculated as follows:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where S(i) is the Silhouette coefficient of the data point i,  
30 a(i) is the average distance between i and all the other data points in the cluster to which i belongs, and

b(i) is the average distance from i to all clusters to which i does not belong.

[0097] The average\_silhouette, e.g., Silhouette score can be then calculated for every K:

$$\text{average\_silhouette} = \text{mean} \{S(i)\}$$

5

[0098] The Silhouette score for each number K can be calculated, where the maximum Silhouette score provides a number K of clusters.

[0099] In an embodiment, the clustering subsystem 204 may perform Silhouette analysis, and determine a number K for the first clusters for the first requestor identifiers and the second clusters for the second requestor identifiers to be 5 each. As a result, the first clusterer 240 clusters the first requestor identifiers into 5 first clusters, and the second clusterer 242 clusters the second requestor identifiers into 5 second clusters.

[0100] FIG. 5 depicts an example of the clustering 500 performed by the data segmentation system 200 according to various embodiments, where a number of clusters is determined to be 5. In an example of FIG. 5, the clustering is depicted with respect to the first requestor identifiers of the first requestor group 230. However, one skilled in the relevant art would understand that the clustering with respect to the second requestor identifiers of the second requestor group 232 may be performed in a similar manner.

[0101] In the clustering 500, e.g., a clustering space, an X-axis represents a percentage of benefit access requests out of a total number of historical access requests, e.g., the second feature, and a Y-axis represents a number of average monthly historical access requests, e.g., the first feature.

[0102] The clustering 500 has five clusters, e.g., a first cluster 510, a second cluster 520, a third cluster 530, a fourth cluster 540, and a fifth cluster 550.

[0103] As shown in FIG. 5, the first requestor identifiers of the first cluster 510 are associated with the greatest number of the average monthly historical access requests, e.g., approximately between 30 and 99, and with a great number of benefit access requests, e.g., approximately over 36%.

[0104] In comparison with the first cluster 510, the first requestor identifiers of the second cluster 520 are associated with a lower number of the average monthly historical access requests, e.g., approximately between 1 and 40, and with a greater number of benefit access requests, e.g., approximately over 64%.

[0105] In comparison with the second cluster 520, the first requestor identifiers of the third cluster 530 are associated with the lower number of the average monthly historical access requests, e.g., approximately between 1 and 36, and with a lower number of benefit access requests, e.g., approximately between 36% and 64%.

- 5 [0106] In comparison with the third cluster 530, the first requestor identifiers of the fourth cluster 540 are associated with a greater number of the average monthly historical access requests, e.g., approximately between 1 and 75, and with a lower number of benefit access requests, e.g., approximately between 1 and 36%.

- 10 [0107] In comparison with the fourth cluster 540, the first requestor identifiers of the fifth cluster 550 are associated with the smallest number of the average monthly historical access requests, e.g., approximately between 1 and 5, and with a greater number of benefit access requests, e.g., approximately between 1 and 44%.

- 15 [0108] Referring again to FIG. 2, as a result of the processing performed by the clustering subsystem 204, the first requestor identifiers of the first requestor group 230 may be grouped into 5 first clusters that define the first segments with respect to the first requestor identifiers. E.g., each first requestor identifier is assigned to one of 5 first clusters, and each first requestor identifier may be associated with at least one from among the first feature, the second feature, the third feature, the fourth feature, and the fifth feature, as described above.

- 20 [0109] Further, the second requestor identifiers of the second requestor group 232 may be grouped into 5 second clusters that define the second segments with respect to the second requestor identifiers. E.g., each second requestor identifier is assigned to one of 5 second clusters, and each second requestor identifier may be associated with at least one from among the first feature, the second feature, the third feature, the fourth feature, and the fifth feature, as described above.

- 25 [0110] The data corresponding to the first clusters, e.g., first training data, and the data corresponding to the second clusters, e.g., second training data, are output as an input for the processing performed by the decision tree learning subsystem 206. The first training data may include information about the first clusters and the first requestor identifiers assigned to the first clusters. The second training data may include information about the second clusters and  
30 the second requestor identifiers assigned to the second clusters.

### *C. Decision Tree*

[0111] The decision tree learning subsystem 206 trains a first decision tree 250 and a second decision tree 252 using the first training data and the second training data, respectively. The training performed by the decision tree learning subsystem 206 is a supervised training, where  
5 the first training data and the second training data provide ground truth with respect to segments into which the first requestor identifiers and the second requestor identifiers are to be segmented into by the trained first decision tree and the trained second decision tree, respectively.

#### **1. First Decision Tree Generation**

10 [0112] In certain implementations, the decision tree learning subsystem 206 includes a first decision tree generator 260. The first decision tree generator 260 receives the first training data corresponding to the first clusters, and performs training using the first training data, to obtain the first decision tree 250 that is capable of segmenting the first requestor identifiers into first segments that match the first clusters. E.g., using the first training data corresponding to the  
15 first clusters, the first decision tree generator 260 derives first rules and thresholds that, when applied on the first requestor identifiers and their associated first requestor features, segment the first requestor identifiers into the first segments that match the first clusters. The first requestor features include, for each first requestor identifier, at least one from among the first feature, the second feature, the third feature, the fourth feature, and the fifth feature, as  
20 described above.

[0113] The first decision tree 250 includes nodes, where each of the nodes stores a first rule associated with a threshold among the first rules associated with various thresholds derived based on the clustering performed by the first clusterer 240, e.g., based on the first training data.

25 [0114] In certain implementations, the data segmentation system 200 may further include a tree criteria determining subsystem 270. The tree criteria determining subsystem 270 includes a Gini impurity calculator 271 and a tree depth adjustor 272. The tree criteria determining subsystem 270 can receive the first training data, and the first rules associated with various thresholds of the first decision tree 250 and its architecture. The Gini impurity calculator 271  
30 calculates Gini impurity index, to determine whether the first rules segment the first requestor identifiers into correct first segments, e.g., the first segments that mirror the first clusters obtained by the first clusterer 240, e.g., based on the first training data. The tree depth adjustor

272 applies a rule-based approach which assigns a maximum number of tree levels that the algorithm could generate. The number of tree levels refers to a tree depth and is a number of nodes from the root node to a leaf node in the longest path. For example, the maximum number of tree levels is adjustable, and may be set to 3. The depth adjustor 272 may apply a rule based  
5 on a maximum number of tree levels, e.g., 3 (see FIG. 6), so that the tree criteria determining subsystem 270 can control the quality of the first decision tree without letting it overgrow. However, the described-above is not limiting. In some embodiments, the maximum number of tree levels may be set to a different number, e.g., 2, 4, 5, etc.

[0115] In some embodiments, based on the Gini impurity index, the first decision tree  
10 generator 260 may then recalculate at least one threshold associated with at least one first rule. As a result, a number of the first requestor identifiers in the first segments received from the first decision tree 250 may better align with the first requestor identifiers in the first clusters. However, this is not intended to be limiting, and, in some implementations, the first decision tree generator 260 does not recalculate any threshold.

15 [0116] When the first rules of the first decision tree 250 is applied on the first requestor identifiers and their associated first requestor features, a segmentation is performed that replicates the clustering performed by the first clusterer 240.

[0117] FIG. 6 depicts a first decision tree according to various embodiments.

[0118] The first decision tree 250 includes a first node 600 to a fifth node 608 that are  
20 arranged in a three level structure. Each of the first node 600 to the fifth node 608 stores an associated rule among the first rules. Each of the first rules describes how to segment the first requestor identifiers with respect to one from among the first feature, the second feature, the third feature, the fourth feature, and the fifth feature and includes a value of a threshold derived by the first decision tree generator 260 using the first training data and/or Gini impurity index.

25 [0119] In an example of the first decision tree 250 depicted in FIG. 6, the main feature for segmenting the first requestor identifiers of the first requestor group 230 is the second feature associated with the benefit access requests, followed by the first feature associated with the average access requests per month and the third feature associated with the portfolio access requests. E.g., in the example, the first rules of the first decision tree 250 segment the first  
30 requestor identifiers based on the first feature, the second feature, and the third feature that are determined based on the clustering results from the first clusterer 240.

5 [0120] The first node 600 is a top node and includes a rule related to the second feature, e.g., the percentage of benefit access requests out of a total number of historical access requests per first requestor identifier. If the percentage of benefit access requests out of a total number of historical access requests per first requestor identifier is greater than a threshold value Th1F2 related to the second feature, then that first requestor identifier is passed on to a second node 602 that is a child node of the first node 600. Otherwise, the first requestor identifier is passed on to a third node 604 that is also a child node of the first node 600. In an embodiment, the threshold value Th1F2 related to the second feature may be 64%, as determined by the first decision tree generator 260.

10 [0121] The second node 602 includes a rule related to the first feature, e.g., monthly average of historical access requests corresponding to the first requestor identifier. If the number of monthly average of historical access requests for the first requestor identifier is greater than a threshold value Th1F1 related to the first feature, then that first requestor identifier is placed into a first bin 610 corresponding to segment 1. Otherwise, the first requestor identifier is  
15 placed into a second bin 612 corresponding to segment 2. In an embodiment, the threshold value Th1F1 related to the first feature may be 33, as determined by the first decision tree generator 260.

[0122] The third node 604 includes another rule related to the second feature, e.g., the percentage of benefit access requests out of a total number of historical access requests per first  
20 requestor identifier. If the percentage of benefit access requests out of a total number of historical access requests per first requestor identifier is greater than a threshold value Th2F2 related to the second feature that is smaller than the threshold value Th1F2, then that first requestor identifier is passed on to a fourth node 606 that is a child node of the third node 604. Otherwise, first requestor identifier is passed on to a fifth node 608 that is also a child node of  
25 the third node 604. In an embodiment, the threshold value Th2F2 related to the second feature may be 36%, as determined by the first decision tree generator 260.

[0123] The fourth node 606 includes another rule related to the first feature, e.g., monthly average of historical access requests per first requestor identifier. If the number of monthly average of historical access requests for the first requestor identifier is greater than a threshold  
30 value Th2F1 related to the first feature that is smaller than Th1F1, then that first requestor identifier is placed into a third bin 620 corresponding to segment 1. Otherwise, the first requestor identifier is placed into a fourth bin 624 corresponding to segment 3. In an

embodiment, the threshold value Th2F1 related to the first feature may be 30, as determined by the first decision tree generator 260.

[0124] The fifth node 608 includes a rule related to the third feature, e.g., the percentage of percentage of portfolio access requests out of a total number of historical access requests per first requestor identifier. If the percentage of portfolio access requests out of a total number of historical access requests per first requestor identifier is greater than a threshold value Th1F3 related to the third feature, then that first requestor identifier is placed into a fourth bin 632 corresponding to segment 5. Otherwise, the first requestor identifier is placed into a fifth bin 634 corresponding to segment 4. In an embodiment, the threshold value Th1F3 related to the third feature may be 31%, as determined by the first decision tree generator 260.

## 2. Second Decision Tree Generation

[0125] With reference again to FIG. 2, in certain implementations, the decision tree learning subsystem 206 includes a second decision tree generator 262. The second decision tree generator 262 receives the second training data corresponding to the second clusters, and performs training using the second training data, to obtain the second decision tree 252 capable of segmenting the second requestor identifiers into second segments that match the second clusters. E.g., using the second training data corresponding to the second clusters, the second decision tree generator 262 derives second rules and thresholds that, when applied on the second requestor identifiers and their associated second requestor features, segment the second requestor identifiers into the second segments that match the second clusters. The second requestor features include, for each requestor identifier, at least one from among the first feature, the second feature, the third feature, the fourth feature, and the fifth feature, as described above.

[0126] The second decision tree 252 includes nodes, where each of the nodes stores a second rule associated with a threshold among the second rules associated with various thresholds derived based on the clustering performed by the second clusterer 242.

[0127] As described above, in certain implementations, the data segmentation system 200 may further include the -Tree Criteria subsystem 270. The Tree Criteria subsystem 270 includes two components, Gini impurity 271 and Tree depth 272. This subsystem 270 can receive the second training data, and the second rules associated with various thresholds of the second decision tree 252 and its architecture. The Gini impurity subsystem 271 calculates Gini

impurity index, to determine whether the second rules segment the second requestor identifiers into correct second segments, e.g., the second segments that mirror the second clusters obtained by the second clusterer 242, e.g., based on the second training data. The Tree depth subsystem 272 applies a rule-based approach which assigns a maximum number of trees that the algorithm could generate. The number of tree levels refers to a tree depth and is a number of nodes from the root node to a leaf node in the longest path. For example, the maximum number of tree levels is adjustable, and may be set to 3. The depth adjustor 272 may apply a rule based on a maximum number of tree levels, e.g., 3 (see FIG. 7), so that the tree criteria determining subsystem 270 can control the quality of the second decision tree without letting it overgrow. However, the described-above is not limiting. In some embodiments, the maximum number of tree levels may be set to a different number, e.g., 2, 4, 5, etc.

**[0128]** In some embodiments, based on the Gini impurity index, the second decision tree generator 262 may then recalculate at least one threshold associated with at least one second rule. As a result, a number of the second requestor identifiers in the second segments received from the second decision tree 252 may better align with the second requestor identifiers in the second clusters. However, this is not intended to be limiting, and, in some implementations, the second decision tree generator 262 does not recalculate any threshold.

**[0129]** When the second rules of the second decision tree 252 are applied on the second requestor identifiers and their associated second requestor features, a segmentation is performed that replicates the clustering performed by the second clusterer 242.

**[0130]** FIG. 7 depicts a second decision tree according to various embodiments.

**[0131]** The second decision tree 252 includes a first node 700 to a sixth node 709 that are arranged in a three level structure. Each of the first node 700 to the sixth node 709 stores an associated second rule among the second rules. Each of the second rules describes how to segment the second requestor identifiers with respect to one from among the first feature, the second feature, the third feature, the fourth feature, and the fifth feature and includes a value of a threshold derived by the second decision tree generator 262 using the second training data and/or Gini impurity index.

**[0132]** In an example of the second decision tree 252 depicted in FIG. 7, the main feature for segmenting the second requestor identifiers of the second requestor group 232 is the third feature associated with the portfolio access requests, where the second feature associated with the benefit access requests, the first feature associated with the average access requests per

month, the fourth feature associated with the non-benefit access requests, and the fifth feature associated with the digitally-engaged access requests are also used. E.g., in the example, the second rules of the second decision tree 252 segment the second requestor identifiers based on the first feature, the second feature, the third feature, the fourth feature, and the fifth feature that are determined based on the clustering results from the second clusterer 242.

**[0133]** The first node 700 is a top node and includes a rule related to the third feature, e.g., the percentage of percentage of portfolio access requests out of a total number of historical access requests per second requestor identifier. If the percentage of portfolio access requests out of a total number of historical access requests per second requestor identifier is greater than a threshold value Th2F3 related to the third feature, then that second requestor identifier is passed on to a second node 702 that is a child node of the first node 700. Otherwise, the second requestor identifier is passed on to a third node 704 that is also a child node of the first node 700. In an embodiment, the threshold value Th2F3 related to the third feature may be 79.7%, as determined by the second decision tree generator 262.

**[0134]** The second node 702 includes another rule related to the third feature, e.g., the percentage of percentage of portfolio access requests out of a total number per second requestor identifier. If the percentage of portfolio access requests out of a total number of historical access requests per second requestor identifier is greater than a threshold value Th3F3 related to the third feature, then that second requestor identifier is passed on to a fourth node 706. Otherwise, the second requestor identifier is placed into an eighth bin 710 corresponding to a segment 10. In an embodiment, the threshold value Th3F3 related to the third feature may be 82.3%, as determined by the second decision tree generator 262.

**[0135]** The fourth node 706 includes a rule related to the fifth feature, e.g., the percentage of digitally-engaged access requests out of a total number of historical access requests per second requestor identifier. If the percentage of digitally-engaged access requests out of a total number of historical access requests per second requestor identifier is greater than a threshold value Th1F5 related to the fifth feature, the second requestor identifier is placed into a seventh bin 712 corresponding to a segment 9. Otherwise, the second requestor identifier is placed into the eighth bin 710 corresponding to the segment 10. In an embodiment, the threshold value Th1F5 related to the fifth feature may be 38.2%, as determined by the second decision tree generator 262.

[0136] The third node 704 includes a rule related to the second feature, e.g., the percentage of benefit access requests out of a total number of historical access requests per second requestor identifier. If the percentage of benefit access requests out of a total number of historical access requests per second requestor identifier is greater than a threshold value Th3F2 related to the second feature, then that second requestor identifier is passed on to a fifth node 708 that is a child node of the third node 704. Otherwise, the second requestor identifier is passed on to a sixth node 709 that is also a child node of the third node 704. In an embodiment, the threshold value Th3F2 related to the second feature may be 40.1%, as determined by the second decision tree generator 262.

[0137] The fifth node 708 includes a rule related to the first feature, e.g., monthly average of historical access requests corresponding to the second requestor identifier. If the number of monthly average of historical access requests for the second requestor identifier is greater than a threshold value Th3F1 related to the first feature, then that second requestor identifier is placed into a tenth bin 724 corresponding to segment 6. Otherwise, the second requestor identifier is placed into an eleventh bin 726 corresponding to a segment 7. In an embodiment, the threshold value Th3F1 related to the first feature may be 7.2, as determined by the second decision tree generator 262.

[0138] The sixth node 709 includes a rule related to the fourth feature, e.g., the percentage of non-benefit access requests out of a total number of historical access requests per second requestor identifier. If the percentage of non-benefit access requests out of a total number of historical access requests per second requestor identifier is greater than a threshold value Th1F4 related to the fourth feature, then that second requestor identifier is placed into a twelfth bin 732 corresponding to a segment 8. Otherwise, the second requestor identifier is placed into a thirteenth bin 734 corresponding to a segment 9. In an embodiment, the threshold value Th1F4 related to the fourth feature may be 52.2%, as determined by the second decision tree generator 262.

### 3. Examples of Segments and Uses

[0139] In embodiments, the segments may be defined separately for the first requestor identifiers and the second requestor identifiers as follows:

First requestor identifiers:  
 Segment 1 - Core requestor  
 Segment 2 - Benefit lover

Segment 3 - All category generalist

Segment 4 - All category digital focus

Segment 5 - Portfolio focused digital

Second requestor identifiers:

5 Segment 6 - Engaged benefit lover

Segment 7 - Benefit lover

Segment 8 - All category digital focus

Segment 9 - Portfolio all category digital focus

Segment 10 - Portfolio digital loyalist

10

**[0140]** The bins shown in FIGS. 6 and 7 illustrate examples of apportionment of samples per segment that was obtained by running an experiment according to the described techniques.

**[0141]** The data obtained by the data segmentation system 200 may be used to develop effective and targeted strategies for the issuers and recommendations for the requestors.

15 However, this is not intended to be limiting. The techniques described herein may be used for other applications for accurate data segmentation based on multiple dimensions and/or where the transmission of the sensitive information from the developer to the customer needs to be minimized or eliminated.

### III. METHODS

20 *A. Decision Tree Training*

**[0142]** As described in detail above, the data segmentation system 200 performs unsupervised ML clustering to segment the access requests into clusters, and then training a rule-based decision tree using the clusters from the clustering, so that the rules of the decision tree, when applied on the access requests, replicate the segmentation as if performed by the  
25 unsupervised ML clustering.

**[0143]** FIG. 8 depicts a flowchart of a method 800 performed by the data segmentation system 200 according to certain embodiments. For example, the method 800 may be performed by some or all of the data sorting subsystem 202, the clustering subsystem 204, the decision tree learning subsystem 206, and the tree criteria determining subsystem 270.

30 **[0144]** The method 800 may be implemented in software (e.g., code, instructions, program) executed by one or more processing units (e.g., processors, cores) of the respective subsystems, using hardware, or combinations thereof. The software may be stored on a non-transitory

storage medium (e.g., on a memory device). The method presented in FIG. 8 and described below is intended to be illustrative and non-limiting. Although FIG. 8 depicts the various processing operations occurring in a particular sequence or order, this is not intended to be limiting. In certain alternative embodiments, the method 800 may be performed in some different order or some operations may be performed in parallel or omitted.

[0145] In 802, the data segmentation system 200 may receive requestor identifiers of a plurality of requestors that are associated with requestor features, as described in detail above.

[0146] In 804, the data segmentation system 200 may perform clustering on some or all of the requestor features, to obtain clusters for the requestor identifiers.

[0147] In 806, the data segmentation system 200 may, using clusters, train a decision tree that segments the requestor identifiers into segments that match the clusters.

*B. Decision Tree Training for Use in Another Computer System*

[0148] As described in detail above, the data segmentation system 200 performs unsupervised ML clustering to segment the access requests into clusters, and then training a rule-based decision tree using the clusters from the clustering, so that the rules of the decision tree, when applied on the access requests, replicate the segmentation as if performed by the unsupervised ML clustering. The rules and the architecture of the decision tree can be then transmitted to another computer system, thereby allowing the other computer system to perform segmentation using access requests for resources from requestors, where the access requests are stored locally on the other computer system.

[0149] FIG. 9 depicts a flowchart of a method 900 performed by the data segmentation system 200 according to certain embodiments. For example, the method 900 may be performed by some or all of the data sorting subsystem 202, the clustering subsystem 204, the decision tree learning subsystem 206, and the tree criteria determining subsystem 270.

[0150] The method 900 may be implemented in software (e.g., code, instructions, program) executed by one or more processing units (e.g., processors, cores) of the respective subsystems, using hardware, or combinations thereof. The software may be stored on a non-transitory storage medium (e.g., on a memory device). The method presented in FIG. 9 and described below is intended to be illustrative and non-limiting. Although FIG. 9 depicts the various processing operations occurring in a particular sequence or order, this is not intended to be limiting. In certain alternative embodiments, the method 900 may be performed in some different order or some operations may be performed in parallel or omitted.

[0151] In 902, the data segmentation system 200 may receive sets of historical access requests for resources from a plurality of requestors, where each historical access request of each of the sets of the historical access requests is associated with a same requestor having a requestor identifier and with a resource category among a plurality of resource categories.

5 Further, each requestor identifier is associated with requestor features. Operation 902 may correspond to operation 802 of FIG. 8.

[0152] In 904, the data segmentation system 200 can cluster requestor identifiers of the plurality of requestors, to obtain a plurality of clusters for the requestor identifiers, by using a plurality of requestor features associated with the requestor identifiers. Operation 904 may

10 correspond to operation 804 of FIG. 8.

[0153] In certain implementations, the data segmentation system 200 clusters the requestor identifiers using K-means clustering.

[0154] In some embodiments, the data segmentation system 200 selects or determines a number K of the plurality of clusters by performing a Silhouette analysis.

15 [0155] In some embodiments, the data segmentation system 200 verifies the segmentation by the application of the decision tree by applying a Gini impurity on the decision tree.

[0156] In 906, the data segmentation system 200 may, using the plurality of clusters, train a decision tree that segments the requestor identifiers into segments that respectively match the plurality of clusters obtained from the clustering. The decision tree includes nodes, each of the nodes storing a rule among a plurality of rules. An application of the decision tree on the plurality of requestor features causes a segmentation being performed that replicates an application of the clustering on requestor identifiers to obtain the clusters. Operation 906 may correspond to operation 806 of FIG. 8.

25 [0157] In 908, the data segmentation system 200 may transmit the rules and an architecture of the decision tree to another computer system, thereby allowing the other computer system to perform segmentation using access requests for resources from requestors, where the access requests are stored locally on the other computer system. The requestors are at least partially the same as the plurality of requestors or are different from the plurality of requestors corresponding to the historical access requests.

30 [0158] In certain implementations, the data segmentation system 200 associates the requestor identifiers with a plurality of requestor groups, respectively, based on resource categories of

each of the sets of historical access requests, where the plurality of requestor groups may include a first requestor group including first requestor identifiers, each of first requestor identifiers being associated with a number of resource categories that is greater than or equal to a first threshold number, respectively, and a second requestor group including second requestor identifiers, each of the second requestor identifiers being associated with a number of resource categories that is smaller than the first threshold number, respectively.

**[0159]** The data segmentation system 200 may perform first clustering using a plurality of first requestor features associated with the first requestor identifiers, to obtain first clusters for the first requestor identifiers, among the plurality of clusters, and second clustering using a plurality of second requestor features associated with the second requestor identifiers, to obtain second clusters for the second requestor identifiers, among the plurality of clusters.

**[0160]** Each of the first clustering and the second clustering may include K-means clustering.

**[0161]** In certain implementations, the data segmentation system 200 selects a first number K of the first clusters by performing a first Silhouette analysis, and selects a second number K of the second clusters by performing a second Silhouette analysis.

**[0162]** The data segmentation system 200 may train, using the first clusters, a first decision tree that segments the first requestor identifiers into first segments that match the first clusters. The first decision tree includes nodes, each of the nodes storing a first rule among a plurality of first rules. An application of the first decision tree on the plurality of first requestor features causes a segmentation being performed that replicates an application of the first clustering on the plurality of first requestor features to obtain the first clusters.

**[0163]** The data segmentation system 200 may also train, using the second clusters, a second decision tree that segments the second requestor identifiers into second segments that match the second clusters. The second decision tree includes nodes, each of the nodes storing a second rule among a plurality of second rules. An application of the second decision tree on the plurality of second requestor features causes a segmentation being performed that replicates an application of the second clustering on the plurality of second requestor features to obtain the second clusters.

[0164] The data segmentation system 200 then may transmit the first rules and an architecture of the first decision tree and the second rules and an architecture of the second decision tree to another computer system, as described above with reference to operation 908.

#### IV. EXAMPLE COMPUTER SYSTEM

5 [0165] Any of the computer systems mentioned herein may utilize any suitable number of subsystems. Examples of such subsystems are in computer system 10 of FIG. 10. In some embodiments, a computer system includes a single computer apparatus, where the subsystems can be the components of the computer apparatus. In other embodiments, a computer system can include multiple computer apparatuses, each being a subsystem, with internal components.

10 A computer system can include desktop and laptop computers, tablets, mobile phones and other mobile devices.

[0166] The subsystems shown in FIG. 10 are interconnected via a system bus 75. Additional subsystems such as a printer 74, keyboard 78, storage device(s) 79, monitor 76 (e.g., a display screen, such as an LED), which is coupled to display adapter 82, and others are shown.

15 Peripherals and input/output (I/O) devices, which couple to I/O controller 71, can be connected to the computer system by any number of means known in the art such as input/output (I/O) port 77 (e.g., USB, FireWire®). For example, I/O port 77 or external interface 81 (e.g. Ethernet, Wi-Fi, etc.) can be used to connect computer system 10 to a wide area network such as the Internet, a mouse input device, or a scanner. The interconnection via system bus 75 allows the

20 central processor 73 to communicate with each subsystem and to control the execution of a plurality of instructions from system memory 72 or the storage device(s) 79 (e.g., a fixed disk, such as a hard drive, or optical disk), as well as the exchange of information between subsystems. The system memory 72 and/or the storage device(s) 79 may embody a computer-readable medium. Another subsystem is a data collection device 85, such as a camera,

25 microphone, accelerometer, and the like. Any of the data mentioned herein can be output from one component to another component and can be output to the user.

[0167] A computer system can include a plurality of the same components or subsystems, e.g., connected together by external interface 81, by an internal interface, or via removable storage devices that can be connected and removed from one component to another component.

30 In some embodiments, computer systems, subsystem, or apparatuses can communicate over a network. In such instances, one computer can be considered a client and another computer a server, where each can be part of a same computer system. A client and a server can each include multiple systems, subsystems, or components.

- [0168] Aspects of embodiments can be implemented in the form of control logic using hardware circuitry (e.g. an application specific integrated circuit or field programmable gate array) and/or using computer software with a generally programmable processor in a modular or integrated manner. As used herein, a processor can include a single-core processor, multi-core processor on a same integrated chip, or multiple processing units on a single circuit board or networked, as well as dedicated hardware. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will know and appreciate other ways and/or methods to implement embodiments of the present invention using hardware and a combination of hardware and software.
- 10 [0169] Any of the software components or functions described in this application may be implemented as software code to be executed by a processor using any suitable computer language such as, for example, Java, C, C++, C#, Objective-C, Swift, or scripting language such as Perl or Python using, for example, any related art or object-oriented techniques. The software code may be stored as a series of instructions or commands on a computer-readable medium for storage and/or transmission. A suitable non-transitory computer-readable medium can include random access memory (RAM), a read only memory (ROM), a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a compact disk (CD) or DVD (digital versatile disk) or Blu-ray disk, flash memory, and the like. The computer-readable medium may be any combination of such storage or transmission devices.
- 15 [0170] Such programs may also be encoded and transmitted using carrier signals adapted for transmission via wired, optical, and/or wireless networks conforming to a variety of protocols, including the Internet. As such, a computer-readable medium may be created using a data signal encoded with such programs. Computer-readable media encoded with the program code may be packaged with a compatible device or provided separately from other devices (e.g., via Internet download). Any such computer-readable medium may reside on or within a single computer product (e.g. a hard drive, a CD, or an entire computer system), and may be present on or within different computer products within a system or network. A computer system may include a monitor, printer, or other suitable display for providing any of the results mentioned herein to a user.
- 20 [0171] Any of the methods described herein may be totally or partially performed with a computer system including one or more processors, which can be configured to perform the steps. Thus, embodiments can be directed to computer systems configured to perform the steps of any of the methods described herein, potentially with different components performing a respective step or a respective group of steps. Although presented as numbered steps, steps of

methods herein can be performed at a same time or at different times or in a different order. Additionally, portions of these steps may be used with portions of other steps from other methods. Also, all or portions of a step may be optional. Additionally, any of the steps of any of the methods can be performed with modules, units, circuits, or other means of a system for performing these steps.

**[0172]** The specific details of particular embodiments may be combined in any suitable manner without departing from the spirit and scope of embodiments of the invention. However, other embodiments of the invention may be directed to specific embodiments relating to each individual aspect, or specific combinations of these individual aspects.

10 **[0173]** The above description of example embodiments of the present disclosure has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the disclosure to the precise form described, and many modifications and variations are possible in light of the teaching above.

15 **[0174]** A recitation of "a", "an" or "the" is intended to mean "one or more" unless specifically indicated to the contrary. The use of "or" is intended to mean an "inclusive or," and not an "exclusive or" unless specifically indicated to the contrary. Reference to a "first" component does not necessarily require that a second component be provided. Moreover, reference to a "first" or a "second" component does not limit the referenced component to a particular location unless expressly stated. The term "based on" is intended to mean "based at least in part on."

20 **[0175]** All patents, patent applications, publications, and descriptions mentioned herein are incorporated by reference in their entirety for all purposes. None is admitted to be prior art.

WHAT IS CLAIMED IS:

1. A method performed by one or more processors of a computer system, the method comprising:

receiving sets of historical access requests for resources from a plurality of requestors, wherein each historical access request of each of the sets of historical access requests is associated with a same requestor having a requestor identifier, each requestor identifier being associated with requestor features;

clustering requestor identifiers of the plurality of requestors to obtain a plurality of clusters for the requestor identifiers, the clustering using a plurality of requestor features associated with requestor identifiers of the plurality of requestors;

training a decision tree that segments the requestor identifiers into segments using the plurality of clusters so that the segments respectively match the plurality of clusters obtained from the clustering, wherein the decision tree comprises nodes, each of the nodes storing a rule among a plurality of rules, wherein an application of the decision tree on the plurality of requestor features causes a segmentation being performed, the segmentation replicating an application of the clustering on the plurality of requestor features to obtain the plurality of clusters; and

transmitting the plurality of rules and an architecture of the decision tree to another computer system, thereby allowing the other computer system to perform segmentation using access requests for resources from requestors, wherein the access requests are stored locally on the other computer system.

2. The method of claim 1, wherein the requestors are at least partially the same as the plurality of requestors or are different from the plurality of requestors that correspond to the sets of historical access requests.

3. The method of claim 1, wherein the clustering comprises K-means clustering.

4. The method of claim 3, wherein the clustering further comprises selecting a number K of the plurality of clusters by performing a Silhouette analysis.

5. The method of claim 1, wherein the training the decision tree further comprises verifying the plurality of rules of the decision tree by applying a Gini impurity on the decision tree.

6. The method of claim 1, wherein each historical access request is associated with a resource category among a plurality of resource categories, and the method further comprises:  
prior to the clustering, associating the requestor identifiers with a plurality of requestor groups, respectively, based on resource categories of each of the sets of historical access requests, the plurality of requestor groups including:

a first requestor group including first requestor identifiers, each of first requestor identifiers being associated with a number of resource categories that is greater than or equal to a first threshold number, respectively, and

a second requestor group including second requestor identifiers, each of the second requestor identifiers being associated with a number of resource categories that is smaller than the first threshold number, respectively.

7. The method of claim 6, wherein the clustering further comprises:  
performing first clustering using a plurality of first requestor features associated with the first requestor identifiers, to obtain first clusters for the first requestor identifiers, among the plurality of clusters; and

performing second clustering using a plurality of second requestor features associated with the second requestor identifiers, to obtain second clusters for the second requestor identifiers, among the plurality of clusters.

8. The method of claim 7, wherein the training the decision tree further comprises:  
training, using the first clusters, a first decision tree that segments the first requestor identifiers into first segments that match the first clusters, wherein the first decision tree comprises nodes, each of the nodes storing a first rule among a plurality of first rules, wherein an application of the first decision tree on the plurality of first requestor features causes a segmentation being performed, the segmentation replicating an application of the first clustering on the plurality of first requestor features to obtain the first clusters; and

training, using the second clusters, a second decision tree that segments the second requestor identifiers into second segments that match the second clusters, wherein the second decision tree comprises nodes, each of the nodes storing a second rule among a plurality of

second rules, wherein an application of the second decision tree on the plurality of second requestor features causes a segmentation being performed, the segmentation replicating an application of the second clustering on the plurality of second requestor features to obtain the second clusters.

9. The method of claim 7, wherein each of the first clustering and the second clustering comprises K-means clustering.

10. The method of claim 9, wherein the first clustering further comprises selecting a first number K of the first clusters by performing a first Silhouette analysis, and the second clustering further comprises selecting a second number K of the second clusters by performing a second Silhouette analysis.

11. A computer system comprising:  
one or more processors; and  
a non-transitory computer-readable storage medium storing one or more instructions that, when executed by the one or more processors, cause the one or more processors to perform a method including:

receiving sets of historical access requests for resources from a plurality of requestors, wherein each historical access request of each of the sets of historical access requests is associated with a same requestor having a requestor identifier, each requestor identifier being associated with requestor features;

clustering requestor identifiers of the plurality of requestors to obtain a plurality of clusters for the requestor identifiers, the clustering using a plurality of requestor features associated with requestor identifiers of the plurality of requestors;

training a decision tree that segments the requestor identifiers into segments using the plurality of clusters so that the segments respectively match the plurality of clusters obtained from the clustering, wherein the decision tree includes nodes, each of the nodes storing a rule among a plurality of rules, wherein an application of the decision tree on the plurality of requestor features causes a segmentation being performed, the segmentation replicating an application of the clustering on the plurality of requestor features to obtain the plurality of clusters; and

transmitting the plurality of rules and an architecture of the decision tree to another computer system, thereby allowing the other computer system to perform segmentation using access requests for resources from requestors, wherein the access requests are stored locally on the other computer system.

12. The computer system of claim 11, wherein the requestors are at least partially the same as the plurality of requestors or are different from the plurality of requestors that correspond to the sets of historical access requests.

13. The computer system of claim 12, wherein the clustering includes K-means clustering.

14. The computer system of claim 11, wherein the clustering further includes selecting a number K of the plurality of clusters by performing a Silhouette analysis.

15. The computer system of claim 11, wherein the training the decision tree further includes verifying the plurality of rules of the decision tree by applying a Gini impurity on the decision tree.

16. The computer system of claim 11, wherein each historical access request is associated with a resource category among a plurality of resource categories, and the method further includes:

prior to the clustering, associating the requestor identifiers with a plurality of requestor groups, respectively, based on resource categories of each of the sets of historical access requests, the plurality of requestor groups including:

a first requestor group including first requestor identifiers, each of first requestor identifiers being associated with a number of resource categories that is greater than or equal to a first threshold number, respectively, and

a second requestor group including second requestor identifiers, each of the second requestor identifiers being associated with a number of resource categories that is smaller than the first threshold number, respectively.

17. The computer system of claim 16, wherein the clustering further includes:

performing first clustering using a plurality of first requestor features associated with the first requestor identifiers, to obtain first clusters for the first requestor identifiers, among the plurality of clusters; and

performing second clustering using a plurality of second requestor features associated with the second requestor identifiers, to obtain second clusters for the second requestor identifiers, among the plurality of clusters.

18. The computer system of claim 17, wherein the training the decision tree further includes:

training, using the first clusters, a first decision tree that segments the first requestor identifiers into first segments that match the first clusters, wherein the first decision tree includes nodes, each of the nodes storing a first rule among a plurality of first rules, wherein an application of the first decision tree on the plurality of first requestor features causes a segmentation being performed, the segmentation replicating an application of the first clustering on the plurality of first requestor features to obtain the first clusters; and

training, using the second clusters, a second decision tree that segments the second requestor identifiers into second segments that match the second clusters, wherein the second decision tree includes nodes, each of the nodes storing a second rule among a plurality of second rules, wherein an application of the second decision tree on the plurality of second requestor features causes a segmentation being performed, the segmentation replicating an application of the second clustering on the plurality of second requestor features to obtain the second clusters.

19. The computer system of claim 17, wherein each of the first clustering and the second clustering includes K-means clustering.

20. The computer system of claim 19, wherein the first clustering further includes selecting a first number K of the first clusters by performing a first Silhouette analysis, and

the second clustering further includes selecting a second number K of the second clusters by performing a second Silhouette analysis.