

learning process based on the set of pipelines. Compared with a pure automatic machine learning process, the hybrid machine learning process consumes less time and/or computing resources as the automatic modeling application 150 can limit its search/optimization to the set of pipelines. Also, by using the user input to create the set of pipelines, it takes advantages of the user's domain knowledge that the automatic modeling application 150 usually does not have.

[0022] The data sources 120 provide electronic data to the machine learning server 110. A data source 120 may be a storage device such as a hard disk drive (HDD) or solid-state drive (SSD), a computer managing and providing access to multiple storage devices, a storage area network (SAN), a database, or a cloud storage system. A data source 120 may also be a computer system that can retrieve data from another source. The data sources 120 may be remote from the machine learning server 110 and provide the data via the network 140. In addition, some or all data sources 120 may be directly coupled to the data analytics system and provide the data without passing the data through the network 140.

[0023] The data provided by the data sources 120 may be organized into data records (e.g., rows). Each data record includes one or more values. For example, a data record provided by a data source 120 may include a series of comma-separated values. The data describe information of relevance to an enterprise using the machine learning server 110. For example, data from a data source 120 can describe computer-based interactions (e.g., click tracking data) with content accessible on websites and/or with applications. As another example, data from a data source 120 can describe customer transactions online and/or in stores. The enterprise can be in one or more of various industries, such as manufacturing, sales, financing, and banking.

[0024] The client devices 130 are one or more computing devices capable of receiving user input as well as transmitting and/or receiving data via the network 140. In one embodiment, a client device 130 is a conventional computer system, such as a desktop or a laptop computer. Alternatively, a client device 130 may be a device having computer functionality, such as a personal digital assistant (PDA), a mobile telephone, a smartphone, or another suitable device. The client devices 130 are configured to communicate with one or more data sources 120 and the machine learning server 110 via the network 140. In one embodiment, a client device 130 executes an application allowing a user of the client device 130 to interact with the machine learning server 110. For example, a client device 130 executes an application to enable interaction between the client device 130 and the machine learning server 110 via the network 140, e.g., through by running a GUI supported by the machine learning server 110. The client

device 130 includes or is otherwise associated with a display device that displays the GUI. The client device 130 is also associated with input devices, e.g., keyboard, mouse, etc., that allow the user to interact with the GUI, such as provide inputs to the GUI. In another embodiment, a client device 130 interacts with the machine learning server 110 through an application programming interface (API) running on a native operating system of the client device 130, such as IOS® or ANDROID™. The client device 130 may interact with one or more data sources 120 to transmit data to a data source 120 or obtain data from a data source 120.

[0025] The network 140 represents the communication pathways between the machine learning server 110 and data sources 120. In one embodiment, the network 140 is the Internet and uses standard communications technologies and/or protocols. Thus, the network 140 can include links using technologies such as Ethernet, 802.11, worldwide interoperability for microwave access (WiMAX), 3G, Long Term Evolution (LTE), digital subscriber line (DSL), asynchronous transfer mode (ATM), InfiniBand, PCI Express Advanced Switching, etc. Similarly, the networking protocols used on the network 140 can include multiprotocol label switching (MPLS), the transmission control protocol/Internet protocol (TCP/IP), the User Datagram Protocol (UDP), the hypertext transport protocol (HTTP), the simple mail transfer protocol (SMTP), the file transfer protocol (FTP), etc.

[0026] The data exchanged over the network 140 can be represented using technologies and/or formats including the hypertext markup language (HTML), the extensible markup language (XML), etc. In addition, all or some of links can be encrypted using conventional encryption technologies such as secure sockets layer (SSL), transport layer security (TLS), virtual private networks (VPNs), Internet Protocol security (IPsec), etc. In another embodiment, the entities can use custom and/or dedicated data communications technologies instead of, or in addition to, the ones described above.

[0027] FIG. 2 is a block diagram illustrating an automatic modeling application 200 according to one embodiment. The automatic modeling application 200 is an embodiment of the automatic modeling application 150 in FIG. 1. The automatic modeling application 200 uses a pipeline to perform an automatic machine learning process on a dataset to train a model. The trained model is to be used for predicting a target variable that may be defined by a user. The automatic modeling application 200 includes a data preparing module 210, a feature engineering module 220, a model training module 230, and a database 240. Those of skill in the art will recognize that other embodiments can have different and/or other components than the ones described here, and that the functionalities can

[0064] FIG. 6 is a high-level block diagram illustrating a functional view of a typical computer system 600 for use as the machine learning server 110 of FIG. 1 according to an embodiment.

[0065] The illustrated computer system includes at least one processor 602 coupled to a chipset 604. The processor 602 can include multiple processor cores on the same die. The chipset 604 includes a memory controller hub 620 and an input/output (I/O) controller hub 622. A memory 606 and a graphics adapter 612 are coupled to the memory controller hub 620 and a display 618 is coupled to the graphics adapter 612. A storage device 608, keyboard 610, pointing device 614, and network adapter 616 may be coupled to the I/O controller hub 622. In some other embodiments, the computer system 600 may have additional, fewer, or different components and the components may be coupled differently. For example, embodiments of the computer system 600 may lack displays and/or keyboards. In addition, the computer system 600 may be instantiated as a rack-mounted blade server or as a cloud server instance in some embodiments.

[0066] The memory 606 holds instructions and data used by the processor 602. In some embodiments, the memory 606 is a random-access memory. The storage device 608 is a non-transitory computer-readable storage medium. The storage device 608 can be a HDD, SSD, or other types of non-transitory computer-readable storage medium. Data processed and analyzed by the machine learning server 110 can be stored in the memory 606 and/or the storage device 608.

[0067] The pointing device 614 may be a mouse, track ball, or other type of pointing device, and is used in combination with the keyboard 610 to input data into the computer system 600. The graphics adapter 612 displays images and other information on the display 618. In some embodiments, the display 618 includes a touch screen capability for receiving user input and selections. The network adapter 616 couples the computer system 600 to the network 140.

[0068] The computer system 600 is adapted to execute computer modules for providing the functionality described herein. As used herein, the term “module” refers to computer program instruction and other logic for providing a specified functionality. A module can be implemented in hardware, firmware, and/or software. A module can include one or more processes, and/or be provided by only part of a process. A module is typically stored on the storage device 608, loaded into the memory 606, and executed by the processor 602.

[0069] The particular naming of the components, capitalization of terms, the attributes, data structures, or any other programming or structural aspect is not mandatory or significant,

We claim:

1. A computer-implemented method, comprising:
receiving a dataset;
generating, automatically and independent of user input, a trained machine learning model using the dataset and an automatic modeling application, the trained machine learning model configured to generate an output based on new data;
displaying, in a user interface of a computing device, a representation of a pipeline that depicts a sequence of decisions made by the automatic modeling application while generating the trained machine learning model using the dataset;
displaying, in the user interface, a plurality of controls, each of the plurality of controls corresponding to at least one decision in the sequence of decisions made by the automatic modeling application while generating the trained machine learning model and configured to receive user input modifying the at least one decision;
receiving, via the user interface, user input to at least one of the plurality of controls modifying one or more decisions in the sequence of decisions; and
causing the automatic modeling application to generate a refined trained machine learning model using a modified sequence of decisions that is defined by the user input to the at least one of the plurality of controls.

2. The computer-implemented method of claim 1, wherein the sequence of decisions includes a data preparation decision and the modified sequence of decisions comprises a data type setting step in the data preparation decision that is different from the sequence of decisions used to generate the trained machine learning model.

3. The computer-implemented method of claim 1, wherein the sequence of decisions includes a data preparation decision and the modified sequence of decisions comprises a data encoding step in the data preparation decision that is different from the sequence of decisions used to generate the trained machine learning model.

4. The computer-implemented method of claim 1, wherein the sequence of decisions includes a data preparation decision and the modified sequence of decisions

comprises a data imputation step in the data preparation decision that is different from the sequence of decisions used to generate the trained machine learning model.

5. The computer-implemented method of claim 1, wherein the sequence of decisions includes a feature engineering decision and the modified sequence of decisions comprises a feature selection step in the feature engineering decision that is different from the sequence of decisions used to generate the trained machine learning model.

6. The computer-implemented method of claim 1, wherein the sequence of decisions includes a feature engineering decision and the modified sequence of decisions comprises a feature ranking step in the feature engineering decision that is different from the sequence of decisions used to generate the trained machine learning model.

7. The computer-implemented method of claim 1, wherein the sequence of decisions includes a model training decision and the modified sequence of decisions comprises a hyperparameter tuning step in the model training decision that is different from the sequence of decisions used to generate the trained machine learning model.

8. The computer-implemented method of claim 1, wherein the sequence of decisions includes a model training decision and the modified sequence of decisions comprises an algorithm selection step in the model training decision that is different from the sequence of decisions used to generate the trained machine learning model.

9. The computer-implemented method of claim 1, wherein the sequence of decisions includes at least one decision that is selected by the trained machine learning model from a plurality of options and the representation of the pipeline includes a ranking score for each of the plurality of options, wherein the ranking score represents a performance measure of the trained machine learning model resulting from selecting an option for the at least one decision.

10. The computer-implemented method of claim 1, wherein the representation of the pipeline includes a feature list comprising a plurality of features and a description, for each of the plurality of features, indicating an importance of the feature relative to the output generated by the trained machine learning model, wherein the plurality of features include features used by the automatic modeling application to generate the trained machine learning model.

11. A system comprising:
one or more processors; and
a computer-readable storage medium storing instructions that are executable by the one or more processors to perform operations comprising:

receiving a dataset;

generating, automatically and independent of user input, a trained machine learning model using the dataset and an automatic modeling application, the trained machine learning model configured to generate an output based on new data;

displaying, in a user interface, a representation of a pipeline that depicts a sequence of decisions made by the automatic modeling application while generating the trained machine learning model using the dataset;

displaying, in the user interface, a plurality of controls, each of the plurality of controls corresponding to at least one decision in the sequence of decisions made by the automatic modeling application while generating the trained machine learning model and configured to receive user input modifying the at least one decision;

receiving, via the user interface, user input to at least one of the plurality of controls modifying one or more decisions in the sequence of decisions; and

causing the automatic modeling application to generate a refined trained machine learning model using a modified sequence of decisions that is defined by the user input to the at least one of the plurality of controls.

12. The system of claim 11, wherein the sequence of decisions includes a data preparation decision and the modified sequence of decisions comprises at least one of a

data type setting step, a data encoding step, or a data imputation step in the data preparation decision that is different from the sequence of decisions used to generate the trained machine learning model.

13. The system of claim 11, wherein the sequence of decisions includes a feature engineering decision and the modified sequence of decisions comprises at least one of a feature selection step or a feature ranking step in the feature engineering decision that is different from the sequence of decisions used to generate the trained machine learning model.

14. The system of claim 11, wherein the sequence of decisions includes a model training decision and the modified sequence of decisions comprises at least one of a hyperparameter tuning step or an algorithm selection step in the model training decision that is different from the sequence of decisions used to generate the trained machine learning model.

15. The system of claim 11, wherein the sequence of decisions includes at least one decision that is selected by the trained machine learning model from a plurality of options and the representation of the pipeline includes a ranking score for each of the plurality of options, wherein the ranking score represents a performance measure of the trained machine learning model resulting from selecting an option for the at least one decision.

16. The system of claim 11, wherein the representation of the pipeline includes a feature list comprising a plurality of features and a description, for each of the plurality of features, indicating an importance of the feature relative to the output generated by the trained machine learning model, wherein the plurality of features include features used by the automatic modeling application to generate the trained machine learning model.

17. A non-transitory computer-readable storage medium storing instructions that are executable by a processing device to perform operations comprising:

receiving a dataset;

generating, automatically and independent of user input, a trained machine learning model using the dataset and an automatic modeling application, the trained machine learning model configured to generate an output based on new data;

displaying, in a user interface, a representation of a pipeline that depicts a sequence of decisions made by the automatic modeling application while generating the trained machine learning model using the dataset;

displaying, in the user interface, a plurality of controls, each of the plurality of controls corresponding to at least one decision in the sequence of decisions made by the automatic modeling application while generating the trained machine learning model and configured to receive user input modifying the at least one decision;

receiving, via the user interface, user input to at least one of the plurality of controls modifying one or more decisions in the sequence of decisions; and

causing the automatic modeling application to generate a refined trained machine learning model using a modified sequence of decisions that is defined by the user input to the at least one of the plurality of controls.

18. The non-transitory computer-readable storage medium of claim 17, wherein the sequence of decisions includes a data preparation decision and the modified sequence of decisions comprises at least one of a data type setting step, a data encoding step, or a data imputation step in the data preparation decision that is different from the sequence of decisions used to generate the trained machine learning model.

19. The non-transitory computer-readable storage medium of claim 17, wherein the sequence of decisions includes a feature engineering decision and the modified sequence of decisions comprises at least one of a feature selection step or a feature ranking step in the feature engineering decision that is different from the sequence of decisions used to generate the trained machine learning model.

20. The non-transitory computer-readable storage medium of claim 17, wherein the sequence of decisions includes a model training decision and the modified sequence

of decisions comprises at least one of a hyperparameter tuning step or an algorithm selection step in the model training decision that is different from the sequence of decisions used to generate the trained machine learning model.