**HUMAN ACTIVITY PATTERNS PREDICTION (HAPP) SYSTEM FOR SMARTHOME APPLICATION**

**Abstract**

This project proposes a model (HAPP) that utilizes smart home big data as a means of learning and discovering human activity patterns for Smart home applications. The proposed model uses frequent pattern mining, cluster analysis, and prediction to measure and analyze energy usage changes sparked by occupants’ behavior. HAPP system addresses the need to analyze energy consumption patterns at the appliance level, which is directly related to human activities. The data from smart meters are recursively mined in the quantum/data slice of 24 h, and the results are maintained across successive mining exercises.

**Existing System:**

* Previous works failed to handle long termbehavior changes.
* Previous approaches have not considered appliance level usage details.
* Operation overlapshas not been considered.
* Deriving accurate predictionshas been a challenge so far.
* Previous models can be used only for short termpredictions.
* Some of the approaches are found to be good at experimental set up but have not consideredreal world scenarios.

The effect of temporal correlationhas not been considered

**Disadvantage:**

* Activities daily living (ADL) of users is monitored and the general activity patterns are modelled according to the user position in his/her environment.
* Thereby, any anomalous or unexpected behaviour of the activity pattern can be detected.
* Moreover, other researches have been employed to mitigate the activity recognition issues with different approaches in various real world activities.
* However, the diversity and complexity in activities are often very high in daily living.

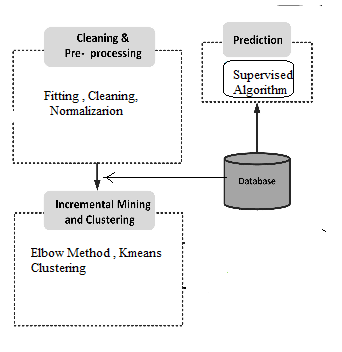
**PROPOSED SYSTEM**

* HAPP model based on appliance usage variations in smart homes appliance.
* This model utilizes FP growth for pattern recognition.
* This model applies k-means clustering algorithm to identify appliance-time associations. This is achieved by Incremental mining.
* A Bayesian network has been applied for activity prediction based on individual and multiple appliance usage.

**Advantage:**

* According to, it was demonstrated that the FP-growth is as successful as other techniques, such as the Eclat method for fast discovery of association rules, a recursive elimination method, Relim, to find frequent item sets and the Decision tree algorithm.
* Therefore, the FP-growth is given as input to the K-pattern clustering and its efficiency contributes widely to the identification of frequent activity patterns of user behavior in the smart home environment.
* The last stage is to group similar patterns by using the frequent activity patterns’ mining.

**System Architecture**



**Literature Survey**

**Author:** **C. Song et al.,**

**Title:** Modeling the scaling properties of human mobility

**Abstract:** While the fat tailed jump size and the waiting time distributions characterizing individual human trajectories strongly suggest the relevance of the continuous time random walk (CTRW) models of human mobility, no one seriously believes that human traces are truly random. Given the importance of human mobility, from epidemic modeling to traffic prediction and urban planning, we need quantitative models that can account for the statistical characteristics of individual human trajectories. Here we use empirical data on human mobility, captured by mobile phone traces, to show that the predictions of the CTRW models are in systematic conflict with the empirical results. We introduce two principles that govern human trajectories, allowing us to build a statistically self-consistent microscopic model for individual human mobility. The model not only accounts for the empirically observed scaling laws but also allows us to analytically predict most of the pertinent scaling exponents.

**Author:** Eunju Kim, Diane Cook, Sumi Helal

# Title: Human Activity Recognition and Pattern Discovery

**Abstract:** In principle, activity recognition can be exploited to great societal benefits, especially in real-life, human centric applications such as elder care and healthcare. This article focused on recognizing simple human activities. Recognizing complex activities remains a challenging and active area of research and the nature of human activities poses different challenges. Human activity understanding encompasses activity recognition and activity pattern discovery. The first focuses on accurate detection of human activities based on a predefined activity model. An activity pattern discovery researcher builds a pervasive system first and then analyzes the sensor data to discover activity patterns.

**Author:** [Junko Murakami](https://ieeexplore.ieee.org/author/37835928900) , [Shin-ichi Ito](https://ieeexplore.ieee.org/author/37287045000)

# Title: Detection of the human-activity using the FCM

**Abstract:** In this paper, we propose the detection system of the human activity by using the electroencephalograms (EEG). First, we measure the EEG data for subjects. In most of all conventional studies, the EEG having a lot of sensors is used. Therefore, subjects must eat or smoke while using the EEG interface. However, this situation is not practical for subjects. In this study, taking account of the burden of subjects, we use only one measurement point 'FPI'. First, we measure the EEG data and the EMG data for subjects. Then, the EEG feature is extracted by using the singular value decomposition (SVD). From the result, we classify the EEG pattern by the fuzzy c-means (FCM). If we cannot classify the EEG pattern into each activity, the discriminant analysis (DA) is used. We consider the EEG features of activities. Then, in order to show the effectiveness of the proposed method, computer simulations are done.

**Author: Ahmed Alsayat, Hoda El-Sayed, Haidong lv,**

**Title: "Efficient Genetic K-Means Clustering for Health Care Knowledge Discovery"**

**Abstract:** Data mining and machine learning are becoming the most interesting research areas and increasingly popular in health organizations. The hidden patterns among patients data can be extracted by applying data mining. The techniques and tools of data mining are very helpful as they provide health care professionals with significant knowledge toward a decision. Researchers have shown several utilities of data mining techniques such as clustering, classification, and regression in health care domain. Particularly, clustering algorithms which help researchers discover new insights by segmenting patients and providing them with effective treatments. This paper, reviews existing methods of clustering and present an efficient K-Means clustering algorithm which uses Self Organizing Map (SOM) method to overcome the problem of finding number of centroids in traditional K-Means. The SOM based clustering is very efficient due to its unsupervised learning and topology preserving properties. Two-staged clustering algorithm uses SOM to produce the prototypes in the first stage and then use those prototypes to create clusters in the second stage. Two health care datasets are used in the proposed experiments and a cluster accuracy metric was applied to evaluate the performance of the algorithm. Our analysis shows that the proposed method is accurate and shows better clustering performance along with valuable insights for each cluster. Our approach is unsupervised, scalable and can be applied to various domains.

**Author: Suwarna Gothane, M. V. Sarode,**

**Title: Analyzing Factors Construction of Dataset Estimating importance of factor and generation of association rules for Indian road Accident**

**Abstract:** Convenience and safety of road is very important and crucial in society. Statistics of accidental deaths has shown a growing drift in India. Accidents on road network with vehicle collision can lead to injuries or even deaths and causes economy loss. Reasons for accident includes over speeding, drunken driver and defects in vehicle as well as bad road condition. Places near residential areas, pedestrian crossing, around schools, college and other educational institutions are the major zone of accidents. Mega cities traffic in India is increasing day by day due to increase in the population. In this paper we have figure out the factor influencing the accidents and identified which factor is more accident prone with Info Gain Attribute Evaluator function using WEKA tool. We have also performed association classification using Apriori algorithm and identified best rule for accident dataset.

**Hardware and Software Requirements:**

**Hardware:**

1. OS – Windows 7,8 or 10 (32 or 64 bit)
2. RAM – 4GB

**Software:**

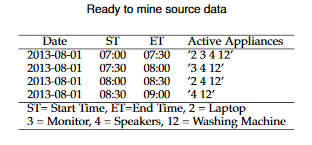
1. Python IDLE
2. Anaconda – Jupiter Notebook

**MODULE:**

1. **Data Preparation**
2. **Extracting Frequent Patterns of Human Activities**
3. **Clustering Analysis: incremental k-means**
4. **Model Selection**

Data Preparation:

he dataset used in this study is a collection of smart meters’ data from five houses in the United Kingdom (UK). This dataset includes 400 million raw records at time resolution of 6 seconds. In the first stage of the cleaning process we developed customized procedures to remove noises from the data and prepare it for mining. After cleaning and preparation, the dataset is reduced to 20 million. Additionally, we developed a synthetic dataset for preliminary evaluation of the model, having over 1.2 million records. In tables (1) and (2) we show an example of the resulting ready to mine source data format comprising four appliances from one house. Smart meters time-series raw data, which is a high time-resolution data, is transformed into a 1-minute resolution load data; subsequently translated into a 30 minutestime-resolution source data, i.e. 24 \* 2 = 48 readings per dayper appliance, while recording start time and end time foreach active appliance.



**Extracting Frequent Patterns of Human Activities**

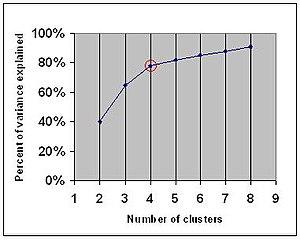
As mentioned earlier, the aim is to discover human activity patterns from smart meters’ data. For example, activities such as” Watching TV, Cooking, Using Computer, Preparing Food and Cleaning Dishes or Clothes” are usually regular routines. Our aim is to detect the patterns of these activities so that a health care application, that monitors sudden changes in patient’s behavior (e.g. patients with cognitive impairment), can send timely alert to health care providers. In pursuing such process, all appliances that are registered active during the 30-minute time interval are included into the source database for frequent pattern data mining. which propose pattern growth or FP-growth approach .

**Clustering Analysis: incremental k-means**

Discovering appliance-to-time associations is vital to health applications that monitor patients’ activity patterns on a daily basis. In this section, a clustering analysis mechanism is used to discover appliance usage time with respect to hour of day (00:00 - 23:59), time of day (Morning, Afternoon, Evening, Night), weekday, week and/or month of the year. Appliance-to-time associations are underlying information in the smart meter time series data which include sufficiently close time-stamps, when relevant appliance has been recorded as active or operational. Using this data, we can group a class or cluster of appliances that are in operation simultaneously or overlapping.

**Elbow Method for choosing k – value**

The **elbow method** is a heuristic method of interpretation and validation of consistency within [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) designed to help find the [appropriate number of clusters in a dataset](https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set). It is often ambiguous and not very reliable, and hence other approaches for [determining the number of clusters](https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set) such as the [silhouette](https://en.wikipedia.org/wiki/Silhouette_(clustering)) method are preferable.



Explained variance. The "elbow" is indicated by the red circle. The number of clusters chosen should therefore be 4.

This method looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion". This "elbow" cannot always be unambiguously identified.[[1]](https://en.wikipedia.org/wiki/Elbow_method_(clustering)#cite_note-1) Percentage of variance explained is the ratio of the between-group variance to the total variance, also known as an [F-test](https://en.wikipedia.org/wiki/F-test). A slight variation of this method plots the curvature of the within group variance

**Model Selection**

To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

**Algoritham :**

**DECISION TREE**

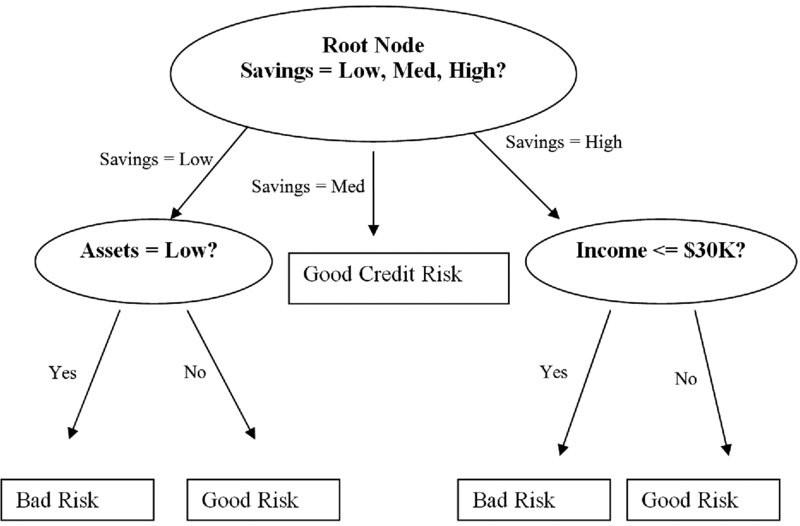
Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**Introduction to Decision Trees:**

A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Decision Tree algorithms are referred to as **CART** **(Classification and Regression Trees)**.



**Common terms used with Decision trees:**

1. **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
4. **Leaf/ Terminal Node:** Nodes do not split is called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
6. **Branch / Sub-Tree:** A sub section of entire tree is called branch or sub-tree.
7. **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the child of parent node.

**Types of Decision Trees**

Types of decision tree is based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** Decision Tree which has categorical target variable then it called as categorical variable decision tree. E.g.:- In above scenario of student problem, where the target variable was “Student will play cricket or not” i.e. YES or NO.
2. **Continuous Variable Decision Tree:** Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

**Advantages of Decision Tree:**

1. **Easy to Understand**: Decision tree output is very easy to understand even for people from non-analytical background. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.
2. **Useful in Data exploration:** Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables / features that has better power to predict target variable. It can also be used in data exploration stage. For e.g., we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable.
3. Decision trees implicitly perform variable screening or feature selection.
4. Decision trees require relatively **little effort from users for data preparation**.
5. **Less data cleaning required:** It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.
6. **Data type is not a constraint:** It can handle both numerical and categorical variables. Can also *handle multi-output problems.*
7. **Non-Parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.
8. Non-linear relationships between parameters do not affect tree performance.
9. The number of hyper-parameters to be tuned is almost null.

**Disadvantages of Decision Tree:**

1. **Over fitting:** Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting. Over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning.
2. **Not fit for continuous variables**: While working with continuous numerical variables, decision tree loses information, when it categorizes variables in different categories.
3. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called **variance**, which needs to be lowered by methods like **bagging** and [**boosting**](https://towardsdatascience.com/boosting-the-accuracy-of-your-machine-learning-models-f878d6a2d185).
4. *Greedy* algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.
5. Decision tree learners create *biased* trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.
6. Information gain in a decision tree with categorical variables gives a biased response for attributes with greater no. of categories.
7. Generally, it gives low prediction accuracy for a dataset as compared to other machine learning algorithms.
8. Calculations can become complex when there are many class label.

**Applications for Decision Tree**:

Decision trees have a natural “if … then … else …” construction that makes it fit easily into a programmatic structure. They also are well suited to categorization problems where attributes or features are systematically checked to determine a final category. For example, a decision tree could be used effectively to determine the species of an animal.

**Algorithm1 Decision Tree Train (data, remaining features)**

1: guess ←most frequent answer in data// default answer for this data

2: if the labels in data are unambiguous then

3: return Leaf(guess)// base case: no need to split further

4: else if remaining features is empty then

5: return Leaf(guess)// base case: cannot split further

6: else // we need to query more features

7: for all f∈ remaining features do

8:NO←the subset of data on which f=no

9: YES ←the subset of data on which f=yes

10: score[f]←# of majority vote answers in NO

11: + # of majority vote answers in YES

// the accuracy we would get if we only queried onf

12: end for

13 f ←the feature with maximal score(f)

14:NO←the subset of data on which f=no

15: YES ←the subset of data on which f=yes

16:left ←Decision Tree Train (NO, remaining features\ {f})

17:right←DecisionTreeTrain (YES, remaining features\ {f})

18:return Node (f,left,right)

19:end if

**K-mean clustering**

It is an unsupervised learning which is used when classlabel is not known or you have unlabeled data. The main focus of this algorithm is finding the groups in the data with that number of groups that represent the variable K. This algorithm iteratively allocating the k groups to the point. Data points here are clustered based on feature of similarity. The consequences of the K-means clustering algorithm are:1)We can use centroid of the Kclusters, to tag new data2)The training data are tagged (A single data point is allocated to a single cluster)Clustering defines groups beforehand observing at the obtainable data, and also allows us to diagnose and examine the groups that have been designed naturally. Each centroid of the accessible clusters is a group of feature ideals that defines the subsequent groups. By studying the centroid eye, weights can easily be used to qualitatively understand that the cluster fits to which group.

**RANDOM FOREST**

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

**How Random Forest Works**

The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

**Advantages Of Using Random Forest**

pros of using random forest for classification and regression.

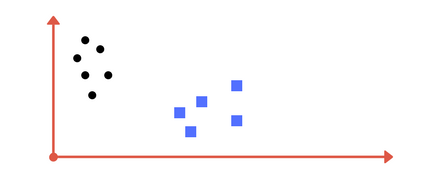
1. The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of "the crowd"; therefore, the overall biasedness of the algorithm is reduced.
2. This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.
3. The random forest algorithm works well when you have both categorical and numerical features.
4. The random forest algorithm also works well when data has missing values or it has not been scaled we.

**SUPPORT VECTOR MACHINE**

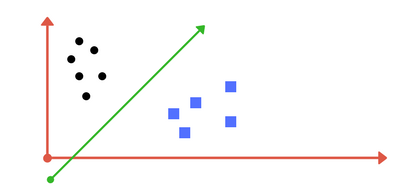
**SUPPORT VECTOR MACHINE**

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

Suppose you are given plot of two label classes on graph as shown in image *(A)*. Can you decide a separating line for the classes?



You might have come up with something similar to following image (*image B*). It fairly separates the two classes. Any point that is left of line falls into black circle class and on right falls into blue square class. *Separation of classes. That’s what SVM does.* It finds out a line/ hyper-plane (in multidimensional space that separate outs classes). Shortly, we shall discuss why I wrote multidimensional space.



# Tuning parameters: Kernel, Regularization, Gamma and Margin.

# **Kernel**

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role.

For **linear kernel** the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

f(x) = B(0) + sum(ai \* (x,xi))

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

SVM constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class, since in general the larger the margin the lower the generalization error of the classifier. The computational load should be sensible, the mappings are used by the SVM scheme to ensure the dot products will be computed in terms of the variable in the original scope, for that a kernel function k(x,y) selected to get the optimal computational time. The higher-dimensional space in the hyper planes is distinct as the set of points whose dot product with a vector in that space is constant. These vectors in the hyper planes defining the hyper planes can be chosen to be linear combinations with parameters of images of feature vectors that occur in the data base.

**DATA FLOW DIAGRAM**

**LEVEL 0**





**LEVEL 1**



**MACHINE LEARNING**

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

**Machine Learning vs. Traditional Programming**

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.



**DATA RULES**

**OUTPUT**

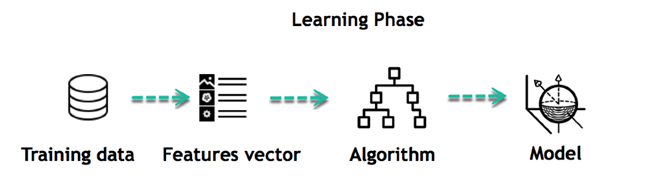
**Machine Learning**

## How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem.

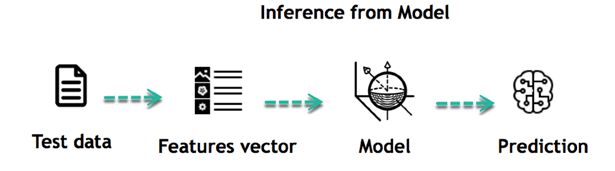
The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.



For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

#### Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

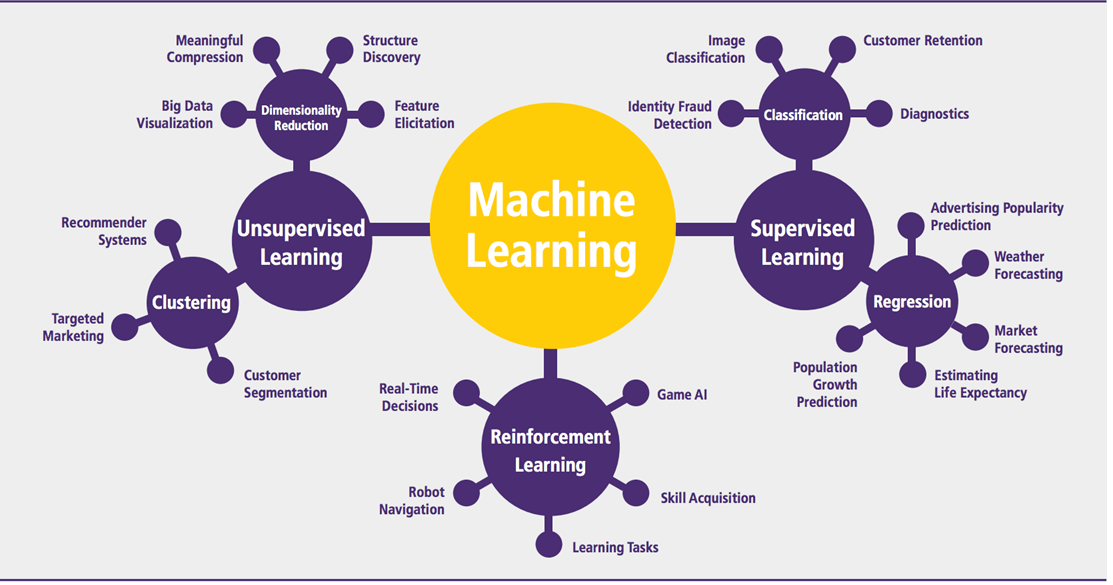


The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

## Machine learning Algorithms and where they are used?



Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

#### Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

* Classification task
* Regression task

#### Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

#### Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

**Machine Learning: Algorithms Types**

Machine learning algorithms are organized into taxonomy, based on the desired outcome of the algorithm. Common algorithm types include:

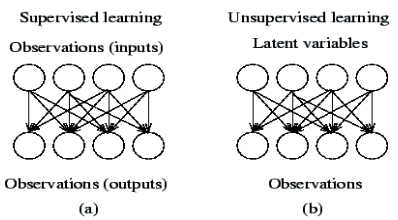
•**Supervised learning** --- where the algorithm generates a function that maps inputs to desired outputs. One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behavior of) a function which maps a vector into one of several classes by looking at several input-output examples of the function.

•**Unsupervised learning** --- which models a set of inputs: labeled examples are not available.

•**Semi-supervised learning** --- which combines both labeled and unlabeled examples to generate an appropriate function or classifier.

•**Reinforcement learning** --- where the algorithm learns a policy of how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback that guides the learning algorithm.

•**Transduction** --- similar to supervised learning, but does not explicitly construct a function: instead, tries to predict new outputs based on training inputs, training outputs, and new inputs. •**Learning to learn** --- where the algorithm learns its own inductive bias based on previous experience. The performance and computational analysis of machine learning algorithms is a branch of statistics known as computational learning theory. Machine learning is about designing algorithms that allow a computer to learn. Learning is not necessarily involving consciousness but learning is a matter of finding statistical regularities or other patterns in the data. Thus, many machine learning algorithms will barely resemble how human might approach a learning task. However, learning algorithms can give insight into the relative difficulty of learning in different environments. Supervised Learning Approach Supervised Learning Supervised learning is fairly common in classification problems because the goal is often to get the computer to learn a classification system that we have created. Digit recognition, once again, is a common example of classification learning. More generally, classification learning is appropriate for any problem where deducing a classification is useful and the classification is easy to determine. In some cases, it might not even be necessary to give pre-determined classifications to every instance of a problem if the agent can work out the classifications for itself. This would be an example of unsupervised learning in a classification context. Examples of Supervised and Unsupervised Learning often leaves the probability for inputs undefined. This model is not needed as long as the inputs are available, but if some of the input values are missing, it is not possible to infer anything about the outputs. Unsupervised learning, all the observations are assumed to be caused by latent variables, that is, the observations is assumed to be at the end of the causal chain. Examples of supervised learning and unsupervised learning are shown in the below:



**Machine Learning Supervise Process**



**Unsupervised learning**

Unsupervised learning[[1]](#footnote-0) seems much harder: the goal is to have the computer learn how to do something that we don't tell it how to do! There are actually two approaches to unsupervised learning. The first approach is to teach the agent not by giving explicit categorizations, but by using some sort of reward system to indicate success. Note that this type of training will generally fit into the decision problem framework because the goal is not to produce a classification but to make decisions that maximize rewards. This approach nicely generalizes to the real world, where agents might be rewarded for doing certain actions and punished for doing others. Often, a form of reinforcement learning can be used for unsupervised learning, where the agent bases its actions on the previous rewards and punishments without necessarily even learning any information about the exact ways that its actions affect the world. In a way, all of this information is unnecessary because by learning a reward function, the agent simply knows what to do without any processing because it knows the exact reward it expects to achieve for each action it could take. This can be extremely beneficial in cases where calculating every possibility is very time consuming (even if all of the transition probabilities between world states were known). On the other hand, it can be very time consuming to learn by, essentially, trial and error. But this kind of learning can be powerful because it assumes no pre-discovered classification of examples. In some cases, for example, our classifications may not be the best possible. One striking example is that the conventional wisdom about the game of backgammon was turned on its head when a series of computer programs (neuro-gammon and TD-gammon) that learned through unsupervised learning became stronger than the best human chess players merely by playing themselves over and over. These programs discovered some principles that surprised the backgammon experts and performed better than backgammon programs trained on pre-classified examples. A second type of unsupervised learning is called clustering. In this type of learning, the goal is not to maximize a utility function, but simply to find similarities in the training data. The assumption is often that the clusters discovered will match reasonably well with an intuitive classification. For instance, clustering individuals based on demographics might result in a clustering of the wealthy in one group and the poor in another. Although the algorithm won't have names to assign to these clusters, it can produce them and then use those clusters to assign new examples into one or the other of the clusters. This is a data-driven approach that can work well when there is sufficient data; for instance, social information filtering algorithms, such as those that Amazon.com use to recommend books, are based on the principle of finding similar groups of people and then assigning new users to groups. In some cases, such as with social information filtering, the information about other members of a cluster (such as what books they read) can be sufficient for the algorithm to produce meaningful results. In other cases, it may be the case that the clusters are merely a useful tool for a human analyst. Unfortunately, even unsupervised learning suffers from the problem of overfitting the training data. There's no silver bullet to avoiding the problem because any algorithm that can learn from its inputs needs to be quite powerful.

**Algorithm Types**

|  |  |  |
| --- | --- | --- |
| **Algorithm Name** | **Description** | **Type** |
| **Linear regression** | Finds a way to correlate each feature to the output to help predict future values. | Regression |
| **Logistic regression** | Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors) | Classification |
| **Decision tree** | Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made | Regression Classification |
| **Naive Bayes** | The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event. | Regression Classification |
| **Support vector machine** | Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver. | Regression (not very common) Classification |
| **Random forest** | The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction. | Regression Classification |
| **AdaBoost** | Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome | Regression Classification |
| **Gradient-boosting trees** | Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it. | Regression Classification |

#### Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Description** | **Type** |
| **K-means clustering** | Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans) | Clustering |
| **Gaussian mixture model** | A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters | Clustering |
| **Hierarchical clustering** | Splits clusters along a hierarchical tree to form a classification system.  Can be used for Cluster loyalty-card customer | Clustering |
| **Recommender system** | Help to define the relevant data for making a recommendation. | Clustering |
| **PCA/T-SNE** | Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances. | Dimension Reduction |

**Application of Machine learning**

**Augmentation**:

* Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

**Automation**:

* Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

**Finance Industry**

* Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

**Government organization**

* The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

**Healthcare industry**

* Healthcare was one of the first industry to use machine learning with image detection.

**Marketing**

* Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

**Example of application of Machine Learning in Supply Chain**

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

**Example of Machine Learning Google Car**

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

**Reinforcement Learning**

Reinforcement learningis a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

* Q-learning
* Deep Q network
* State-Action-Reward-State-Action (SARSA)
* Deep Deterministic Policy Gradient (DDPG)

**Applications/ Examples of deep learning applications**

**AI in Finance:**The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.

Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

**AI in HR:**Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

**AI in Marketing:**AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers.

For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

**Artificial Intelligence**

## Difference between Machine Learning and Deep Learning

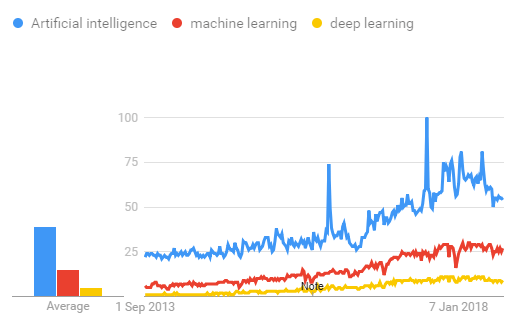
|  |  |  |
| --- | --- | --- |
|  | **Machine Learning** | **Deep Learning** |
| **Data Dependencies** | Excellent performances on a small/medium dataset | Excellent performance on a big dataset |
| **Hardware dependencies** | Work on a low-end machine. | Requires powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication |
| **Feature engineering** | Need to understand the features that represent the data | No need to understand the best feature that represents the data |
| **Execution time** | From few minutes to hours | Up to weeks. Neural Network needs to compute a significant number of weights |
| **Interpretability** | Some algorithms are easy to interpret (logistic, decision tree), some are almost impossible (SVM, XGBoost) | Difficult to impossible |

## When to use ML or DL?

In the table below, we summarize the difference between machine learning and deep learning.

|  |  |  |
| --- | --- | --- |
|  | **Machine learning** | **Deep learning** |
| **Training dataset** | Small | Large |
| **Choose features** | Yes | No |
| **Number of algorithms** | Many | Few |
| **Training time** | Short | Long |

With machine learning, you need fewer data to train the algorithm than deep learning. Deep learning requires an extensive and diverse set of data to identify the underlying structure. Besides, machine learning provides a faster-trained model. Most advanced deep learning architecture can take days to a week to train. The advantage of deep learning over machine learning is it is highly accurate. You do not need to understand what features are the best representation of the data; the neural network learned how to select critical features. In machine learning, you need to choose for yourself what features to include in the model.



## TensorFlow

the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

* Researchers
* Data scientists
* Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

In this tutorial, you will learn

**TensorFlow Architecture**

Tensor flow architecture works in three parts:

* Preprocessing the data
* Build the model
* Train and estimate the model

It is called Tensor flow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

**Where can Tensor flow run?**

TensorFlow can hardware, and software requirements can be classified into

Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.

Run Phase or Inference Phase: Once training is done Tensor flow can be run on many different platforms. You can run it on

* Desktop running Windows, macOS or Linux
* Cloud as a web service
* Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of TensorFlow is the Tensor Board. The Tensor Board enables to monitor graphically and visually what TensorFlow is doing.

**List of Prominent Algorithms supported by TensorFlow**

* Linear regression: tf. estimator. Linear Regressor
* Classification: tf. estimator. Linear Classifier
* Deep learning classification: tf. estimator. DNN Classifier
* Deep learning wipe and deep: tf. estimator. DNN Linear Combined Classifier
* Booster tree regression: tf. estimator. Boosted Trees Regressor
* Boosted tree classification: tf. estimator. Boosted Trees Classifier

**DESIGN**

**UML diagrams**

The unified modeling is a standard language for specifying, visualizing, constructing and documenting the system and its components is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure and control information about the systems. Depending on the development culture, some of these artifacts are treated more or less formally than others. Such artifacts are not only the deliverables of a project; they are also critical in controlling, measuring, and communicating about a system during its development and after its deployment. The UML addresses the documentation of a system&#39;s architecture and all of its details. The UML also provides a language for expressing requirements and for tests. Finally, the UML provides a language for modeling the activities of project planning and release management.

Building blocks of UML:

The vocabulary of the UML encompasses three kinds of building blocks:

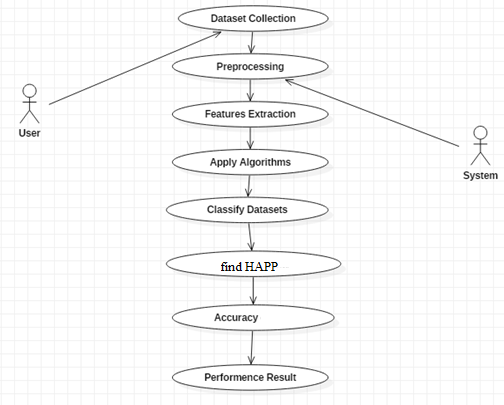
* Things
* Relationships
* Diagrams

Things in UML:

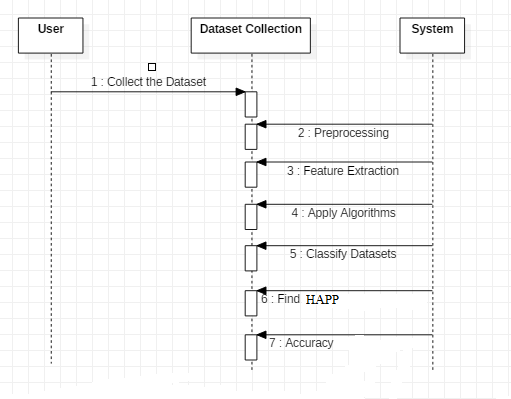
There are four kinds of things in the UML:

* Structural things
* Behavioural things
* Grouping things
* Annotational things

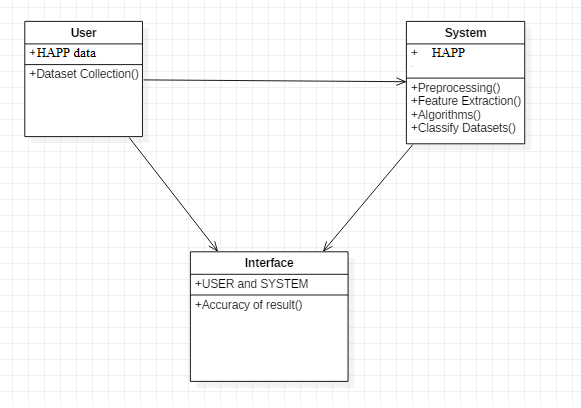
**Use Case Diagram:**



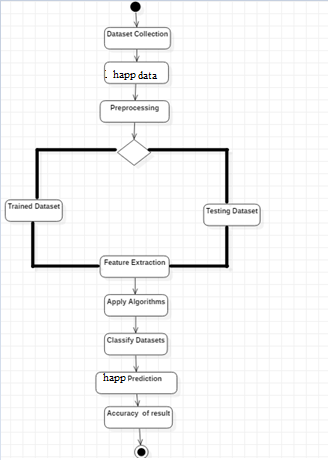
**Sequence Diagram:**



**Class Diagram:**



**Activity Diagram:**



**TESTING**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

* Meets the business and technical requirements that guided its design and Development.
* Works as expected and can be implemented with the same characteristics.

**TESTING METHODS**

* **Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* Functions: Identified functions must be exercised.
* Output: Identified classes of software outputs must be exercised.
* Systems/Procedures: system should work properly

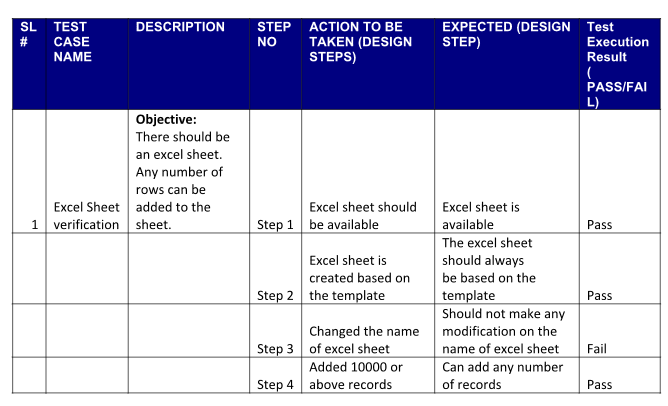
**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

Test Case for Excel Sheet Verification:

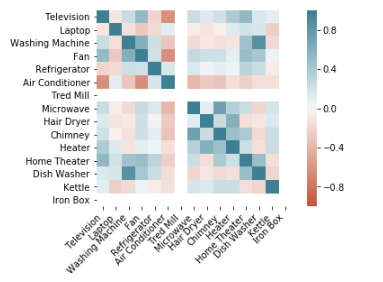
Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on classification will work on the respective columns of dataset .

Test Case 1 :



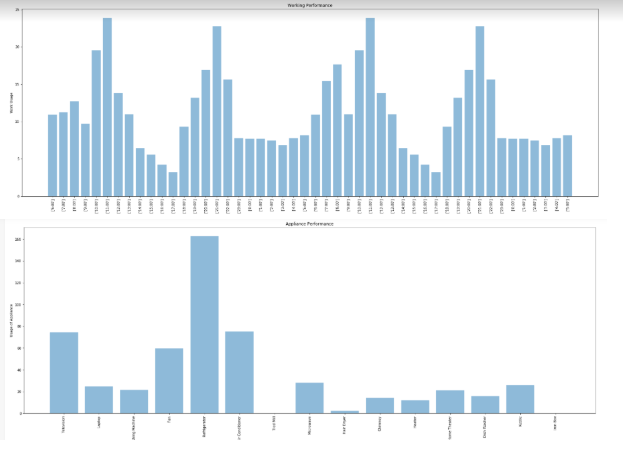
**Results:**

**Correlation Matrix**

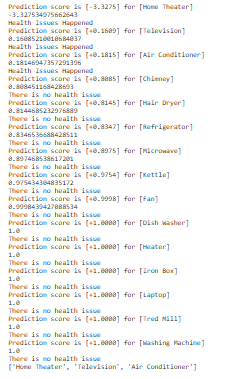


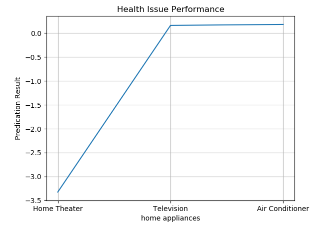
**EDA**

**Data Visualization**

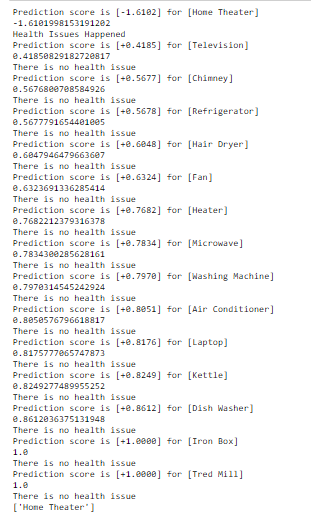


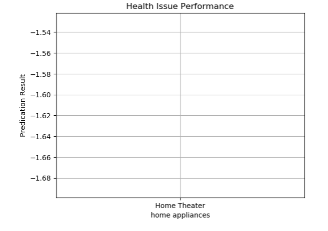
**Predication Result DecisionTree Regressor**



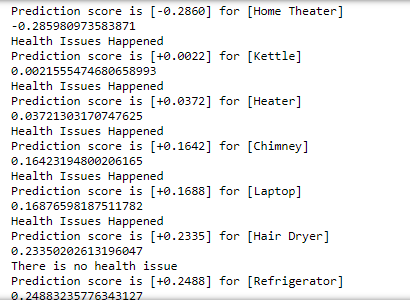


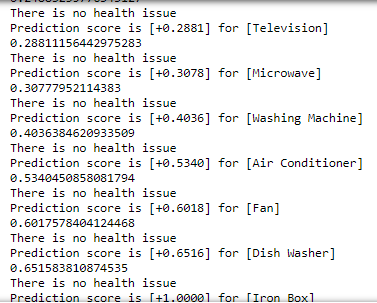
**Predication Result RandomForest Regressor**

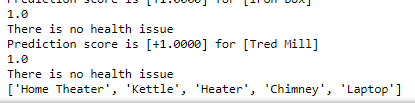


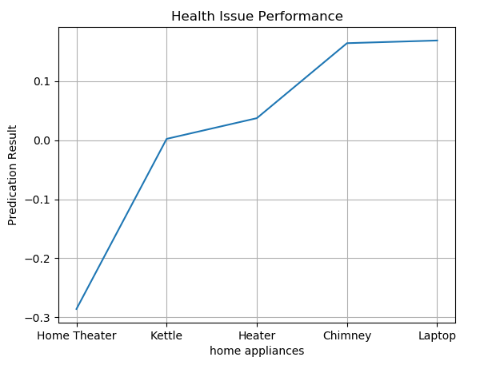


**Predication Result SVM Regressor**









**Conclusion**

The proposed model uses frequent pattern mining, cluster analysis, and prediction to measure and analyze energy usage changes sparked by occupants’ behavior. HAPP system addresses the need to analyze energy consumption patterns at the appliance level, which is directly related to human activities. We monitor pattern using FP-growth algorithm and predicting using K-means clustering and decision tree.

**Future Work**

For future work, we are planning to refine the model and introduce distributed learning of big data mining from multiple houses in a near real-time manner. This will help health applications to promptly take actions such as sending alerts to patients or care providers. Furthermore, we are planning to build a health ontology model to automatically map discovered appliances to potential activities. This means we can efficiently train the system and increase the accuracy of detecting human activities.

**References**

[1] H. Kalantarian, C. Sideris, B. Mortazavi, Dynamic computation ofﬂoading for low-power wearable health monitoring systems, IEEE Transactions on Biomedical Engineering, 64(3): 621–628, 2017.

[2] S.H. Ahme d, D. Kim, Named data networking-based smart home, Ict Express, 2(3): 130–134, 2016.

[3] S.S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: asurvey, Artiﬁcial Intelligence Review, 43(1): 1–54, 2015.

[4] F.J. Ord´ o˜nez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, Sensors, 16(1): 115, 2016.

[5] D. Cook, KD. Feuz, NC. Krishnan, Transfer learning for activity recognition: Asurvey , Knowledge and Information Systems, 36(3): 537–556, 2013.

[6] L. Wang, Recognition of human activities using continuous autoencoders with wearable sensors, Sensors, 16(2): 189, 2016.

[7] O.D.Lara,MA.Labrador,Asurveyonhumanactivityrecognitionusingwearablesensors,IEEECommunicationsSurveys and Tutorials, 15(3): 1192–1209, 2013.

[8] Y. Bengio, Deep learning of representations: Looking forward, in International Conference on Statistical Language and Speech Processing, 1–37, 2013.

[9] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, ACM Computing Surveys (CSUR), 46(3): 33, 2014.

[10] R. Chavarriaga, H. Sagha, A. Calatroni, The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition, Pattern Recognition Letters, 34(15): 2033–2042, 2013.

[11] D. Anguita, A. Ghio, L. Oneto, A public domain dataset for humanactivityrecognitionusingsmartphones,ESANN,2013.

[12] J. Wang, Y. Chen, S. Hao, Deep learning for sensor-based activity recognition: A survey, Pattern Recognition Letters, 2018

[13] J. Mantyjarvi, J. Himberg, T. Seppanen, Recognizing human motion with multiple acceleration sensors, in 2001 IEEE International Conference on Systems, Man and Cybernetics, 2: 747–752, 2001.

[14] D.O. Olguın, AS. Pentland, J. Fahrenberg, Human activity recognition: Accuracyacrosscommonlocationsforwearable sensors,Proceedingsof200610thIEEEInternationalSymposiumonWearableComputers,Montreux,Switzerland,11–14, 2006.

[15] J.R. Kwapisz, GM. Weiss, SA. Moore, Activity recognition using cell phone accelerometers, ACM SigKDD Explorations Newsletter, 12(2): 74–82, 2011.

[16] C.A.Ronao,SB.Cho,Humanactivityrecognitionwithsmartphone sensors using deep learning neural networks, Expert Systems with Applications, 59: 235–244, 2016.

[17] N.Y. Hammerla, S. Halloran, T. Ploetz, Convolutional, and recurrent models for human activity recognition using wearables, arXiv preprint arXiv:1604.08880, 2016.

[18] O.Banos,JM.Galvez,M.Damas,Windowsizeimpactinhuman activity recognition, Sensors, 14(4): 6474–6499, 2014.

[19] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural

Computation, 9(8): 1735–1780, 1997.

[20] J.K. Chorowski, D. Bahdanau, D. Serdyuk, Attention-based models for speech recognition, Advances in Neural Information Processing Systems, 577–585, 2015.

[21] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473, 2014.

[22] K.M. Hermann, T. Kocisky, E. Grefenstette, Teaching machines to read and comprehend, Advances in Neural Information Processing Systems, 1693–1701, 2015.

[23] Z. Lin, M. Feng, CN. Santos, A structured self-attentive sentence embedding, arXiv preprint arXiv:1703.03130, 2017.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, Dropout: a simple way to prevent neural networks from overﬁtting, Journal of Machine Learning Research, 15(1): 1929–1958, 2014.

[25] R. Pascanu, T. Mikolov, Y. Bengio, Understanding the exploding gradient problem, CoRR, abs/1211.5063, 2012.

1. [↑](#footnote-ref-0)