VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**LÊ THANH VINH - 520H0597**

**NGUYỄN TIẾN ĐẠT - 520H0527**

**NGUYỄN HOÀNG PHÚC - 521H0509**

**FINAL REPORT**

**INTRODUCTION TO MACHINE LEARNING**

**HO CHI MINH CITY, YEAR 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**LÊ THANH VINH - 520H0597**

**NGUYỄN TIẾN ĐẠT - 520H0527**

**NGUYỄN HOÀNG PHÚC - 521H0509**

**FINAL REPORT**

**INTRODUCTION TO MACHINE LEARNING**

*Instructor*:

**Mr. Lê Anh Cường**

**HO CHI MINH CITY, YEAR 2023**

**GRATITUDE**

Dear Ton Duc Thang University administration and Faculty of Information Technology,

We are writing to express my sincere gratitude for the opportunity to complete the midterm report.

Through this experience, we have gained valuable knowledge and skills that have helped us to advance our professional development. We are particularly grateful for the guidance and support of our professors.

We would like to thank you again for your support and encouragement. We are committed to continuing my education and becoming a valuable member of society.

*TP. Hồ Chí Minh, ngày 22 tháng 12 năm 2023*

*Tác giả*

*Lê Thanh Vinh*

*Nguyễn Tiến Đạt*

*Nguyễn Hoàng Phúc*

**THE PROJECT WAS COMPLETED**

**AT TON DUC THANG UNIVERSITY**

I hereby declare that this is my own research work, under the scientific guidance of Mr. Le Anh Cuong. The research contents and results of this thesis are honest and have not been published in any form before. The data in the tables and graphs used for analysis, comments, and evaluation are collected by the author from various sources, as indicated in the reference section.

In addition, the Project also uses some comments, evaluations, and data from other authors and organizations, all of which are cited and referenced.

**If any fraud is detected, I will be solely responsible for the content of my Project.** Ton Duc Thang University is not responsible for any copyright infringements caused by me during the implementation process (if any).

*City. Ho Chi Minh, date 22. month 12 year 20.*

*Author*

*Lê Thanh Vinh*

*Nguyễn Tiến Đạt*

*Nguyễn Hoàng Phúc*

# DATA ANALYSIS

## Introduction to data

ObesityDataSet\_raw\_and\_data\_sinthetic is a data set used to classify a person's obesity level based on lifestyle factors and health conditions. This data includes information collected from diverse sources, including hospitals and scientific research. This dataset is highly diverse, allowing for extensive analysis of the association between variables and obesity, and provides a reliable database for building prediction models and research in the field. health field. This dataset contains 2112 samples, each representing one person. Each pattern has 17 features, including:

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature Name** | **Description** | **Category** | **Variable Type** |
| Gender | Gender of each person | Responder Characteristics | Categorical |
| Age | The person's age | Responder Characteristics | Integer |
| Height | The person's height in meters | Responder Characteristics | Float |
| Weight | The person's weight is measured in kg | Responder Characteristics | Float |
| family\_history\_with\_overweight | Whether or not a family member is obese | Responder Characteristics | Categorical |
| FAVC | Does the person frequently eat high-calorie foods? | Eating Habits | Categorical |
| FCVC | Frequency of consumption of vegetables | Eating Habits | Ordinal |
| NCP | Number of main meals | Eating Habits | Ordinal |
| CAEC | Consumption of food between meals | Eating Habits | Ordinal |
| SMOKE | Smokes Yes or No | Physical Conditioning | Categorical |
| CH2O | Daily water consumption level | Eating Habits | Ordinal |
| SCC | Whether to monitor calorie consumption or not | Physical Conditioning | Categorical |
| FAF | Physical activity frequency | Physical Conditioning | Ordinal |
| TUE | Time using technology devices | Physical Conditioning | Ordinal |
| CALC | Frequency of alcohol consumption | Eating Habits | Ordinal |
| MTRANS | Transportation used | Physical Conditioning | Categorical |
| NObeyesdad | Level of obesity | Target Variable | Categorical |

The target variable is NObeyesdad. The attempt is is apply Machine Learning to find the best model for predicting NObeyesdad. NObeyesdad is a categorical variable that is a measure of a person weight ranging from under weight to very obese.

## Understand the data

In this section, we will analyze, explain and find evidence to find out how the features will affect the prediction results based on actual knowledge. Features are classified into 3 main groups including: eating habits, physical conditioning and responder characteristics for easier analysis.

### Eating Habits

The energy value of food is measured in units called calories. An average physically active man needs about 2,500 calories per day to maintain a healthy weight, and an average physically active woman needs about 2,000 calories per day. If this calorie consumption threshold is exceeded without physical activity to consume calories, it will be the main cause of obesity, so eating habits greatly affect whether that person is obese or not. The features of this group including: FAVC, FCVC, NCP, CAEC, CH20 and CALC.

FAVC: Feature that represents the frequency of eating foods that contain a lot of calories, and as mentioned above, excess calories are a cause of obesity because this can be an important feature that greatly affects the prediction. Therefore, this feature may very useful in making predictions.

FCVC: Feature representing the frequency of vegetable consumption, vegetables are a group of foods that are very low in calories and are rich in fiber, which makes people feel full quickly and limits the consumption of foods high in calories, so the frequency of vegetable consumption will also have a significant impact on the prediction results. Therefore, this feature may useful in making predictions.

NCP: This feature represents the number of main meals in a day. As mentioned, obesity is caused by excessive calorie intake. The number of main meals in a day cannot indicate whether the calorie intake is high or low that day because a person may have few main meals but each meal contains a lot of calories or there may also be many main meals but each meal contains very few calories. Therefore, this feature may not be useful in making predictions.

CAEC: This feature represents the frequency of food consumption outside of main meals. This can affect the amount of calories that the body intakes. Therefore, this feature may very useful in making predictions.

CH20: A feature that represents the frequency of water consumption per day. According to an experimental study taking place at Boston Children's Hospital, Boston, Massachusetts, from February 2011 to June 2014 observed weight changes between two groups of subjects who drank a lot of water and less water. The results showed that there was not much difference in weight between these two groups. Therefore, this feature may not be useful in making predictions.

CALC: This feature represents the frequency of alcohol consumption. Alcohol is a high-calorie beverage, so drinking too much alcohol can lead to obesity. One study has shown that alcohol consumption may contribute to excess energy consumption. associated with weight gain in some individuals. Therefore, this feature may useful in making predictions.

### Physical Conditioning

Lack of physical activity is another important factor associated with obesity. If you are not active enough, you will not use the energy provided by the foods you eat, and the excess energy you consume will be stored by your body as fat. So physical activities are also important features to make predictions. The features of this group including: SCC, FAF, TUE, MTRANS and SMOKE.

SCC: This is a feature that represents whether or not you need to track your daily calorie intake. It can have a big impact on calorie intake because when we monitor our calorie intake, we can control it. So it also has an impact on whether the person is obese or not. Therefore, this feature may useful in making predictions.

FAF: A feature that represents whether a person regularly performs physical activities or not. This is one of the important things that determines whether a person is obese or not. Therefore, this feature may very useful in making predictions.

TUE: A feature that represents the frequency of using technology devices. According to a new Nielsen Company survey, Americans are spending more time than ever looking at screens. During the first quarter of 2016, U.S. people spent an average of 10 hours and 39 minutes per day using devices such as tablets, cellphones, personal computers, video games, and televisions, an hour higher than the previous year. Sedentary activities, such as screen time, have been related to an increased risk of obesity, which raises the risk of diabetes. Obesity and diabetes rates have both risen in the United States in recent years. Therefore, this feature may very useful in making predictions.

MTRANS: A feature that represents what type of transportation is used. Researchers from the University of Illinois at Urbana-Champaign and Georgia Tech conducted a new study that compared and evaluated county data from 2001 to 2009.

They discovered that in counties across the United States, a one percentage point increase in mass transit use is associated with a 0.473 percentage point reduced obesity rate. It can be seen that the type of transportation used will also affect obesity. Therefore, this feature may useful in making predictions.

SMOKE: A feature that represents whether the person smokes or not. There is research showing that current smokers are less likely to be obese than people who have never smoked. Former smokers are more likely to be obese than both current and never smokers. Among smokers, the risk of obesity increases with the number of cigarettes smoked, and former heavy smokers are more likely to be obese than light former smokers. The risk of obesity gradually decreases with time of quitting smoking. After 30 years, former smokers still have a higher risk of obesity than current smokers but the risk is equivalent to people who have never smoked. It can be said that smoking has a great influence on a person's likelihood of becoming obese. Therefore, this feature may useful in making predictions.

### Responder Characteristics

Responder Characteristics is also an important part of deciding whether a person is obese or not. Indicators such as height and weight will also say something. The features of this group including: family\_history\_with\_overweight, Gender, Age, Height and Weight.

family\_history\_with\_overweight: A feature that represents whether a person's family history is obese or not. Based on a practical study, it has been shown that family history of obesity and cardiometabolic diseases are important risk factors for precocious obesity onset in childhood and are related to the severity of obesity. Therefore, this feature may useful in making predictions.

Gender: A feature that represents the person's gender. According to a study by the Indian Longitudinal Study of Aging. The analytical sample size for the study was 28,050 adults aged 60 years and older. Obesity based on body mass index (BMI) is more common in older women than in men (26.3% vs. 17.6%). The study's findings suggest significant gender differences in the prevalence of multimorbidity (including obesity). Therefore, this feature may useful in making predictions.

Age: a feature that represents the person's gender. According to statistics, obesity rates increase sharply with age from puberty to late middle age. In 2001, 25.6% of Americans aged 60 to 69 were obese, compared with only 14% of Americans aged 18 to 29. Therefore, this feature may useful in making predictions.

Height and Weight: are 2 features used to calculate BMI which can conclude whether the person is obese or not, so these 2 features greatly affect the prediction results. Therefore, this feature very useful in making predictions.

## Analyze and evaluate data

After analyzing to determine how the features will affect the prediction results based on real-world knowledge, this section will proceed to analyze based on the results in the ObesityDataSet\_raw\_and\_data\_synthetic dataset by plotting a graph for each feature against the target variable to examine the correlation between each feature and the target.

# Some methods to improve the accuracy of machine learning models

## Hyperparameter Tuning

Hyperparameters are parameters that cannot be directly learned from the training process with data. They are typically set before the actual training process begins. Hyperparameters are often used to tune model performance, and they can have a significant impact on a model's accuracy, generality, and other metrics.

Hyperparameter tuning is the process of choosing optimal values for the hyperparameters of a machine learning model. The goal of hyperparameter tuning is to find the values that lead to the best performance for a given task. In each different model there will be different hyperparameters such as hyperparameters in Neural Networks, Support Vector Machine and so on. There are several techniques for hyperparameter tuning such as:

GridSearchCV: This techniques used to find the best set of hyperparameters for the model from a predetermined list of values. It performs testing of all possible combinations of hyperparameters to determine the most optimal combination for the performance of the prediction model. This process is done through creating a grid of possible values for each hyperparameter and performing model training and evaluation with each of these combinations to find the best set of hyperparameters.

RandomizedSearchCV: This technique performs the random selection of a fixed number of hyperparameter combinations from a specified distribution. Instead of going through every combination, RandomizedSearchCV randomly selects a number of hyperparameter sets to test. Each iteration, RandomizedSearchCV tries a different set of hyperparameters and records the model's performance. It returns the combination that gives the best results after many iterations. This can be more efficient for large hyperparameter spaces, saving time and computational resources.

Bayesian Optimization: This technique uses information from previous trials to find the next points in the hyperparameter space that are likely to yield the best results. Instead of trying each value one by one, it uses a probabilistic model to predict the potential locations of the best hyperparameter sets based on information gathered from previous experiments.

## Ensemble Learning

Ensemble Learning is a machine learning technique that improves prediction accuracy by merging predictions from multiple models into a final prediction by considering multiple perspectives and using the strengths of different models, Ensemble Learning improves the overall performance of the learning system. Some popular techniques in Ensemble Learning include: Bagging, Boosting, Stacking and so on.

### Bagging

Bagging is combining the results of multiple models to get a general result. By building a large number of models on different subsamples from the training dataset. These models will be trained independently and in parallel, but their output will be averaged to produce the final result. When creating all the models on the same dataset and combining it, there is a high chance that the models will give the same results because they receive the same input data. This problem can be solved using bootstrapping technique.

Bootstrapping is a technique that uses an existing data sample to create many different hypothetical data samples through resampling. During the resampling process, subsets of observations from the original data set with replacement are created. By repeating the resampling process many times, bootstrapping creates many different hypothetical data samples.

Some popular Bagging models and algorithms include Random Forests, Bagging meta-estimator and so on.

### Boosting

Boosting is a technique of combining weak models to create a stronger model by building models sequentially in a way that focuses on correcting the errors of the previous model. More specifically, Boosting builds a large number of models, each of which will learn how to correct the errors of the previous model. The weight updated through each model of correctly predicted data will remain unchanged, while the weight of incorrectly predicted data will be increased to focus more on them in building the next model. Then create a new model that is better able to classify previously erroneous data points. There are several Boosting algorithms included:

AdaBoost (Adaptive Boosting): AdaBoost gives an equal weight to every data set. The algorithm then automatically adjusts the weights of the data points after each decision tree. This algorithm gives greater weight to misclassified items to fix these for the next round. The algorithm repeats the process until the residual error or difference between the actual value and the predicted value is lower than the acceptance threshold.

Gradient Boosting Machines (GBM): GBM does not give greater weight to misclassified items. Instead, GBM software optimizes the loss function by creating machine learning bases in sequence, the gradient of the loss function is calculated to know the direction and magnitude to be adjusted to minimize error and then Then a new weak model is built to predict the missing part of the residual error, so the current base learning machine is always more effective than the previous learning machine.

### Stacking

Build several models (usually of different types) and a meta model (supervisor model), train these models independently, then the meta model will learn how to best combine the prediction results of several models. Stacking typically considers heterogeneous weak learners, then parallelizes them and combines them by training a meta-learner to make predictions based on the weak learners' different predictions. The algorithm takes the output of the submodels as input and tries to learn how to best combine the input predictions to make better output predictions.

## Validation

Overfitting is also a cause leading to models with low accuracy; to mitigate this, there are numerous methods, specifically here, one can utilize the technique of using validation sets.

During model training, a portion of the training data is reserved as a validation set. The model is trained on the remainder of the training data and evaluated on the validation set. This process is repeated multiple times while tuning the model or hyperparameters to achieve better performance. The validation set serves as an independent data set to evaluate how well the model generalizes to new, never-before-seen data. This validation set helps detect problems such as overfitting or underfitting. The validation set helps tune hyperparameters, select the best model, and prevent overfitting. By evaluating the performance of the model on the validation set, one can tune the structure of the model or the training process.

There is another variation of validation sets, that is cross-validation. This is a technique where the training data is divided into multiple subsets, and the model is trained and evaluated on each subset. Cross-validation is useful when the dataset is not too large. In validation, it may require up to 20% of the training data to be used as a validation set. In cross-validation, it may only need 10% in each subset to be used as a validation set because it is evaluated multiple times.

## Feature Engineering

Prediction models include an outcome variable and predictor variables, and during the feature engineering process, useful predictor variables are created and selected for the prediction model. Feature engineering involves creating, transforming, extracting, and selecting the features, also known as variables, that are most beneficial for creating accurate Machine Learning algorithms.

### Feature Engineering Implementation

#### Feature Creation

Creating features entails identifying the valuable variables for a predictive model. It's a subjective task that demands human intervention and creativity. Current features are blended by employing addition, subtraction, multiplication, and scaling to craft novel derived features possessing superior predictive capabilities. Data is compared to a canvas, and feature creation to the brushstrokes that form a captivating artwork. The process entails more than just gathering raw data; it also entails developing features that reveal hidden patterns and insights. Combining features, creating polynomial representations, and capturing temporal aspects through time-based features are examples of brushstrokes that bring out the nuances in data.

#### Transformation

In the realm of data, transformations play a crucial role in optimizing features for machine learning algorithms. Scaling and normalization ensure harmonized feature scales, facilitating understanding by models like k-means clustering and support vector machines. Logarithmic or exponential transformations refine data symmetry, aligning it with specific model characteristics. Binning reshapes continuous variables into categorical harmonies, capturing intricate relationships. Overall, transformations involve manipulating predictor variables to enhance model flexibility, accuracy, and performance, creating a symphony of features that contribute to improved algorithmic analysis and understanding. Guarantees the model's adaptability to diverse datasets, ensures uniformity in variable scales, simplifying model comprehension, enhancing accuracy, and preventing computational errors by maintaining all features within the model's acceptable range.

#### Feature Extraction

In the context of high-dimensional data, feature extraction acts as a librarian in a vast library of books, selecting the most insightful volumes to present to the model. This process involves the automatic creation of new variables from raw data, aiming to reduce data volume into a more manageable set for modeling. Tools like Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (tSNE), and Autoencoders serve as powerful instruments, distilling knowledge and reducing dimensions without sacrificing contextual information. Feature extraction methods encompass various techniques, including cluster analysis, text analytics, edge detection algorithms, and principal components analysis, contributing to the automatic generation of informative variables for improved modeling efficiency.

#### Feature Selection

Feature selection algorithms essentially analyze, evaluate and rank different features to determine which features are irrelevant and need to be eliminated, which features are redundant and need to be eliminated, removed as well as which features are most useful to the model and should be prioritized.

### Feature engineering techniques

#### Imputation

Imputation can be envisioned as the artistic process of filling in the blanks on your data canvas. In the world of data analysis, missing values can pose challenges for many algorithms. Imputation techniques, such as using the mean, median, or more advanced methods like k-nearest neighbors and regression imputation, act as brushes that delicately restore completeness to the dataset.

Imputation methods play a vital role in ensuring that datasets are filled with meaningful values. These techniques are particularly essential when dealing with algorithms that may struggle or produce biased results in the presence of missing data. In the analogy, imputation serves as the artistic touch that brings the entire data canvas to life by addressing the absence of certain values.

#### Handling Outliers

Handling outliers is a crucial aspect of data analysis to ensure the robustness and accuracy of statistical models. Outliers are data points that deviate significantly from the majority of the dataset and can potentially skew the results. Here are common techniques for handling outliers.

Removing Outliers: This involves excluding data points that are identified as outliers. While this can be effective, it should be done judiciously, as removing too many outliers may lead to a loss of valuable information.

Winsorizing: Winsorizing is a method that involves capping extreme values, replacing them with values closer to the mean or within a specified range. This technique helps control the influence of outliers without completely eliminating them.

Interquartile Range (IQR) Method: The Interquartile Range method involves defining a range based on the interquartile range, which is the difference between the third quartile (Q3) and the first quartile (Q1) of the dataset. Data points outside this range are considered outliers and can be treated accordingly.

#### Log Transform.

The logarithmic transformation serves as a tool for addressing the imbalance and skewed nature of data. Imagine data as a seesaw, sometimes lopsided and affecting your perceptions. The logarithmic transformation acts as a stabilizing force by compressing large values and expanding small ones, bringing equilibrium to the data.

If the data exhibits exponential growth or has long tails, the logarithmic transformation helps align it with the modeling rhythm. Much like adjusting the balance of a seesaw, the logarithmic transformation ensures that extreme values are moderated, allowing for a more even representation. This transformation is especially valuable when dealing with data that shows significant variations in magnitude, as it brings a sense of balance and consistency to the modeling process.

#### Feature Split

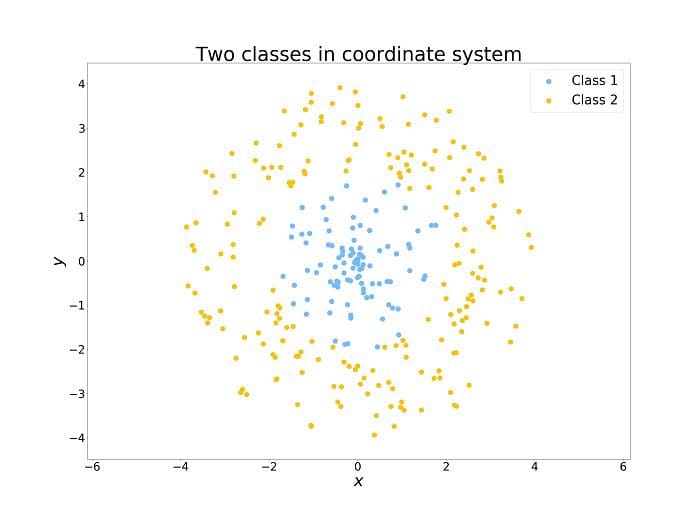
Feature splitting, especially in the context of text data, serves as a linguistic translator for machine learning models. In the realm of text, where words come together to form narratives, feature splitting takes a sentence and dissects it into individual words or n-grams. This process is akin to a translator breaking down a linguistic expression into its fundamental components.

Imagine you have a sentence such as "The quick brown fox." Through feature splitting, this sentence is transformed into individual components: "The," "quick," "brown," and "fox." Additionally, it can involve creating n-grams, which are sequences of adjacent words like "The quick," "quick brown," and "brown fox."

### Feature Engineering Example

Feature engineering involves modifying or creating features in a dataset to enhance the performance of machine learning algorithms. The text provides a simple example to illustrate the concept.

Example Scenario: Two classes of points are represented on a plot. The task is to supply clients from a warehouse located at a specific point, and profitability is constrained by a limited distance from the warehouse.



Human vs Algorithmic Perspective: Humans easily understand the need to consider points within a limited radius. However, algorithms, particularly decision tree-based ones, analyze features one at a time and make splits based on arbitrary thresholds. Dividing the space into limited radii would require numerous splits.

Feature Engineering Solution: To address this, the example introduces a feature engineering technique: transforming Cartesian coordinates (x, y) to polar coordinates (r, θ). The transformation simplifies the analysis for both humans and algorithms. The Transformation is defined as :

# 

And now, it is simple for a human to see, as well as for an algorithm to analyze the data. At the threshold r split = 2, divide the set with a split along r. Obviously, this is a simplified example, and real-world data is rarely that simple, but it demonstrates the power of proper feature engineering for machine learning.

# IMG_256

## Data Augmentation

The process of artificially expanding a dataset by generating new data points from existing ones is known as data augmentation. This is accomplished through techniques such as making minor changes to the data or using deep learning models to generate additional data points. Essentially, it is a method of increasing the training set by producing modified copies of the dataset using the existing data, either through slight alterations or the generation of new data through deep learning approaches.

### Augmented vs Synthetic data

Augmented data involves making minor changes to the original dataset, particularly in image augmentation, where geometric and color space transformations are applied (e.g., flipping, resizing, cropping, adjusting brightness and contrast) to enhance the training set's size and diversity. On the other hand, synthetic data is artificially generated without using the original dataset and often employs Deep Neural Networks (DNNs) and Generative Adversarial Networks (GANs). It's worth noting that augmentation techniques are not limited to images; they can be applied to various data types such as audio, video, and text. Synthetic data generation is just one method of data augmentation, with other approaches including making minimal changes to existing data to create new data points.

### Some method to apply Data Augmentation

Padding: Padding involves adding extra pixels around the borders of an image. It helps prevent information loss during subsequent transformations and ensures that the model learns features from the entire image.

Random Rotating: Randomly rotating an image introduces variability and helps the model become invariant to different orientations. This is particularly useful for tasks where the orientation of the object is not crucial.

Re-scaling: Rescaling involves changing the size of the image. This can be done by zooming in or out to provide the model with different perspectives of the same object.

Vertical and Horizontal Flipping: Flipping an image horizontally or vertically introduces mirror images of the original, helping the model become invariant to the orientation of objects.

Translation: Translation involves shifting an image along the X and Y directions. This helps the model learn to recognize objects regardless of their position in the image.

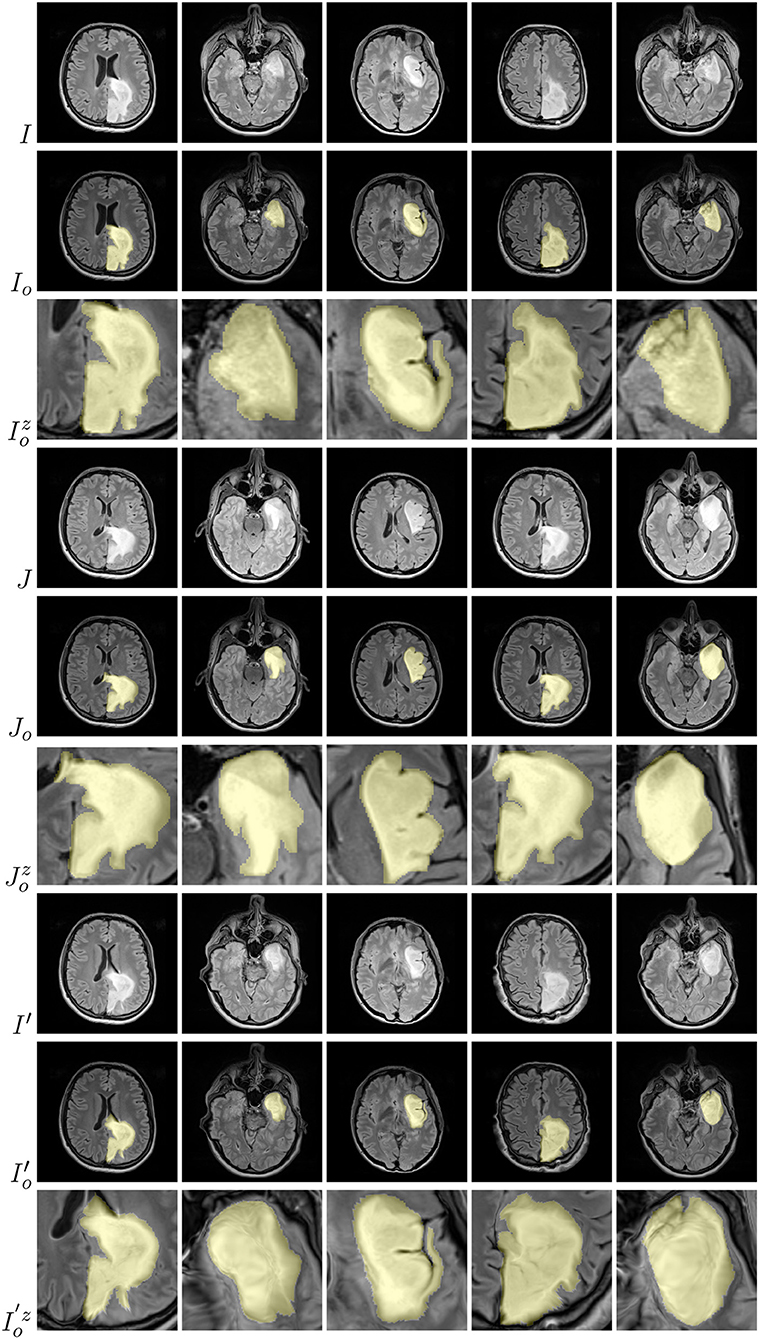
Cropping: Cropping involves removing parts of the image. This can help the model focus on specific features and learn to recognize objects in various contexts.

Zooming: Zooming in or out changes the scale of the image, allowing the model to learn to recognize objects at different distances.

### Example of Data Augmentation

Data augmentation can be applied to any machine learning application where obtaining good-quality data is a challenge. Furthermore, it can help improve model robustness and performance across all fields of study.

An example about healthcare: To avoid overfitting, deep models require a large amount of ground-truth data. Obtaining high-quality ground-truth data for medical images, particularly brain-tumor delineation from MRI, is time-consuming, costly, and human-dependent. Furthermore, annotated datasets are frequently unbalanced, with some classes under-represented. Data augmentation techniques are being actively developed to address the scarcity of medical training sets. These techniques generate synthetic training examples, which aid in mitigating overfitting and improving deep models' generalization capabilities.



The use of diffeomorphic image registration on example brain images resulted in visually plausible generated images. We also show tumor masks overlayed over the corresponding original images (in yellow; rows with the o subscript), as well as a zoomed part of a tumor (rows with the z superscript) for source (I), target (J), and artificially generated (I′) images.

An example about Self-Driving Cars: Accurate 3D object identification is critical for self-driving vehicle safety, and LiDAR sensors play an important role in this task by generating high-definition point clouds. 3D Object Detection models then use these point clouds to identify and detect objects. However, training effective Object Detection models necessitates a large amount of data, which is both expensive and time-consuming to collect and process. Data Augmentation is required to maximize the utility of each data sample. Data Augmentation is especially important for improving performance, and is frequently regarded as equally important as advancements in the Object Detection models themselves.

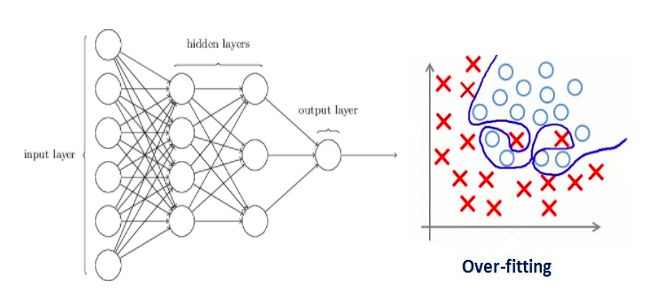


## Regularization Techniques

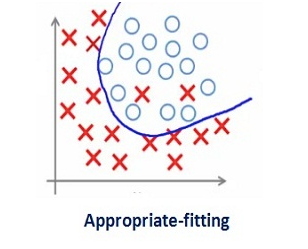
Regularization in machine learning and deep learning is a crucial technique aimed at preventing overfitting and enhancing the generalization performance of models. It involves adding a penalty term to the loss function during training, discouraging the model from becoming overly complex or having large parameter values. Regularization methods, including L1 and L2 regularization, dropout, and early stopping, contribute to making models more robust and improving their accuracy on unseen data.

In the context of machine learning, regularization refers to a set of strategies that go beyond simple memorization to help models generalize more effectively. Overfitting occurs when a model memorizes the training data but performs poorly on unseen data. Regularization techniques reduce or regularize feature coefficients, preventing overfitting and promoting a better balance of learning and memorization. The ultimate goal is to create models that perform consistently across both training and test datasets, thereby addressing the issues associated with overfitting.

### How does Regularization help reduce Overfitting?



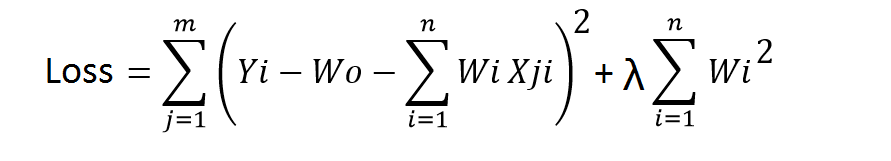
Regularization in machine learning is known for penalizing coefficients. However, in the context of deep learning, the emphasis shifts to penalizing the weight matrices associated with the nodes.Assume that our regularization coefficient is so large that some of the weight matrices are close to zero. As a result, the linear network will be much simpler and the training data will be slightly underfitted. Because a large amount of regularization coefficient is not required, the best solution is to optimize regularization coefficient to obtain a well-fitted model.



### Different Regularization Techniques

#### Ridge Regression (L2 Regularization)

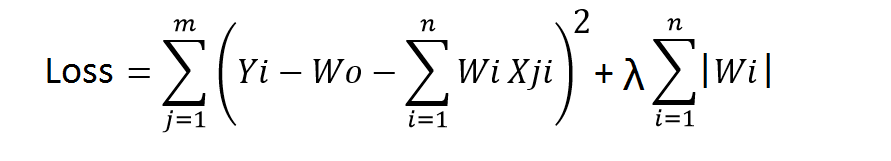
In the context of L2 regularization, the original loss function is adjusted by introducing a term that involves the sum of squared weights. This addition is a way to normalize the weights in the model.



The parameter λ is a crucial factor in this process and is fine-tuned using a cross-validation dataset. When λ is set to 0, the loss function reverts to the initial choice, which might be the residual sum of squares. On the other hand, for a very high value of λ, the loss function gives more importance to minimizing the sum of squared weights than the core loss function. In extreme cases, it might lead the model to set the parameters values to zero.

Essentially, the learning of parameters is now influenced by this modified loss function. The objective becomes minimizing this function, encouraging the parameters to be as small as possible. The introduction of the L2 regularization term, with its dependence on λ, acts as a preventive measure, restraining the weights from becoming excessively large and contributing to a more controlled and generalized model by favoring smaller weights.

#### Lasso Regression (L1 Regularization)



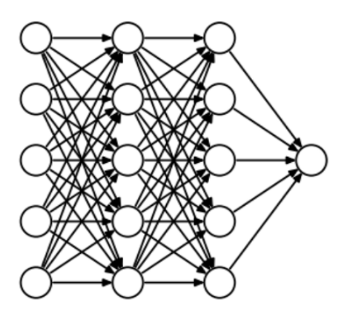
In contrast to Ridge Regression, Lasso regularization takes a different approach by using the absolute values of weights for normalization. Like in Ridge Regression ,λ remains a tuning parameter with a similar role.

The distinctive feature of Lasso is that it incorporates only the absolute values of weights in the loss function. Consequently, during the optimization process, algorithms penalize higher weight values. While Ridge Regression tends to bring parameters close to zero without exactly reaching zero, Lasso, on the other hand, takes a more decisive step. It not only regularizes the model but also performs feature selection by setting the weights of less important features precisely to zero.

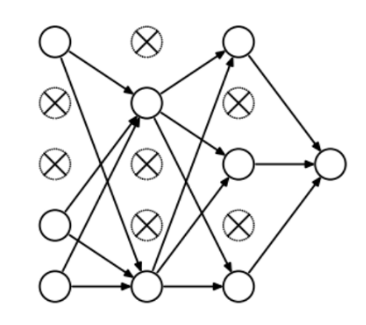
In essence, Lasso regularization goes beyond just controlling the magnitude of weights; it actively prunes less influential features from the model, effectively performing regularization and feature selection simultaneously. This characteristic makes Lasso a valuable tool when aiming for a sparser and more interpretable model by excluding irrelevant features.

#### Dropout

Dropout, a popular regularization technique in deep learning, particularly in neural networks, is well known for its ability to produce robust results. The essence of dropout is the random selection and temporary removal of nodes and their associated connections during each iteration of the training process.



Dropout introduces a stochastic element to the framework of a neural network, which is represented as interconnected nodes, by systematically eliminating a subset of nodes in each iteration. This dynamic exclusion produces a variety of outputs over iterations, similar to the principles of ensemble techniques in machine learning. Ensemble models, which use diverse subsets of data, outperform individual models by capturing more variability—a principle that dropout also applies to neural networks. When there is no more Overfitting, the model's accuracy will also improve



The probability of selecting and dropping nodes functions as a hyperparameter governing the dropout technique. This stochastic process is not confined to hidden layers but extends to input layers as well, injecting randomness to curb overfitting.