UNIVERSIDAD
Panamericana

MACHINE LEARNING: THE PRACTICAL PATH

COM 139: SIMULATION & VISUALIZATION [1]

1 INTRODUCTION

Machine learning is a subfield of artificial intelligence that focuses on developing algorithms and systems that can learn from data and make decisions or predictions without being explicitly programmed for every scenario. At its core, machine learning enables computers to identify patterns in large datasets and use those patterns to make informed choices. This capability has transformed numerous industries, from healthcare and finance to entertainment and transportation, by enabling smarter systems and automation.

The process of machine learning typically involves feeding a model with data, allowing it to learn relationships within that data through a training process. Models can be categorized into different types depending on the nature of the task—such as supervised learning for labeled data, unsupervised learning for discovering hidden patterns, and reinforcement learning for sequential decision-making. Common algorithms include linear regression, decision trees, support vector machines, and neural networks, each suited to different types of problems and data structures.

As data becomes increasingly central to decision-making, machine learning provides a powerful toolkit for extracting insights, automating tasks, and solving complex problems. However, the effectiveness of a machine learning system depends not only on the algorithm used but also on the quality of the data, the choice of features, and the interpretation of results. Understanding these fundamentals is key to building reliable, ethical, and impactful machine learning solutions.

2 WHAT PROBLEMS CAN IT SOLVE?

ML is a powerful tool used to solve a wide variety of problems by learning patterns from data and making predictions or decisions without being explicitly programmed for each specific task. The problems that ML can address are often categorized based on the nature of the learning process and the type of output they produce. Broadly, these include **supervised learning**, **unsupervised learning**, and **reinforcement learning** problems. Each type is suited to different scenarios, ranging from predicting outcomes based on labeled data to discovering hidden patterns or optimizing decisions through trial and error.

Among the most common *supervised learning* problems are **classification** and **regression**. In *classification*, the goal is to assign data into predefined categories—for example, determining whether an email is spam or not, or identifying objects in images. *Regression*, on the other hand, deals with predicting continuous values, such as forecasting house prices or stock market trends based on various input features. These problems are tackled using labeled datasets where the algorithm learns to map inputs to correct outputs, making them highly effective for real-world applications in business, healthcare, finance, and more.

Unsupervised learning, which includes problems like **clustering** and **dimensionality reduction**, is used when the data lacks explicit labels. *Clustering algorithms*, for example, can group similar data points together—such as customer segmentation based on purchasing behavior—without prior knowledge of the

[1] *Engeneering Faculty, Universidad Panamericana, Guadalajara, México*

group categories. *Reinforcement learning*, a third type, is well-suited for **decision-making problems** where an agent learns to perform actions in an environment to maximize a reward, like in robotics or game playing. Together, these types of problems showcase the breadth of challenges machine learning can address, offering flexible solutions across domains where traditional programming would be too rigid or complex.

For this project we are going to focus our efforts in **regression** and **classification** problems.

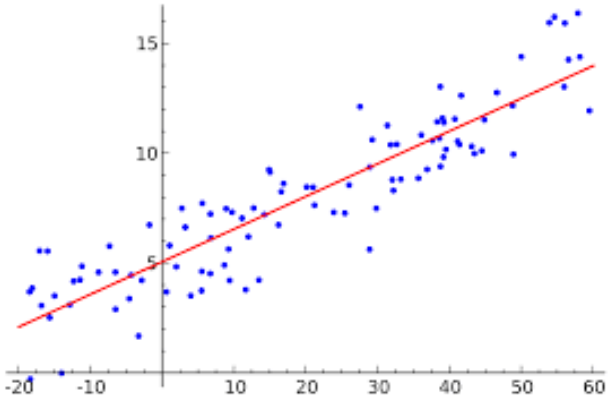
3 REGRESSION ALGORITHMS

Regression algorithms in machine learning are designed to model the relationship between a set of input variables (features) and a continuous output variable. Unlike classification, where the output is a category or class label, regression aims to predict real-valued outcomes—such as prices, temperatures, or probabilities. These algorithms learn patterns from historical data to estimate or forecast numeric values based on new inputs. Common examples of regression tasks include predicting house prices from property features, estimating sales revenue over time, or forecasting stock prices.

There are various types of regression algorithms, each with its own strengths and assumptions. Linear regression is the simplest and most widely used, assuming a linear relationship between inputs and output. More flexible models like polynomial regression, decision tree regression, and random forest regression can capture complex, nonlinear patterns. The choice of algorithm depends on the nature of the data, the distribution of the target variable, and the desired level of interpretability versus performance. Evaluating regression models typically involves metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), or R-squared, which help determine how closely the predictions align with actual values.

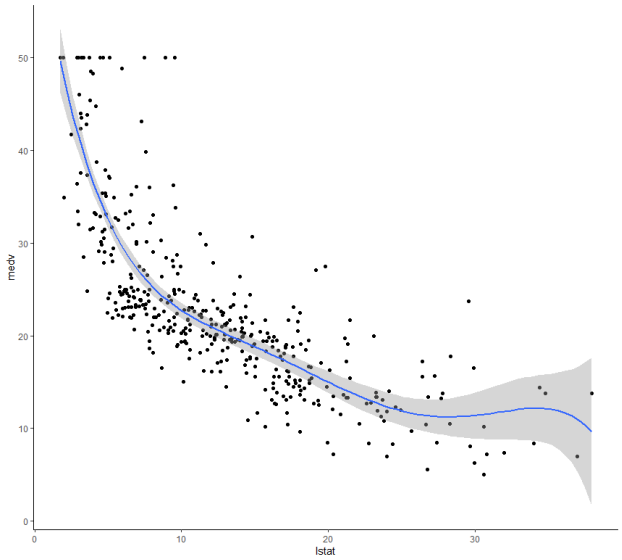
3.1 Linear Regression

Linear regression is a technique used to model the relationships between observed variables. The idea behind simple linear regression is to "fit" the observations of two variables into a linear relationship between them. Graphically, the task is to draw the line that is "best-fitting" or "closest" to the points (x_i, y_i), where x_i and y_i are observations of the two variables which are expected to depend linearly on each other.



3.2 Polynomial Regression

Polynomial regression is an extension of linear regression where the relationship between the input variables and the output is modeled as an nth-degree polynomial. It allows the model to fit nonlinear patterns in the data by adding powers of the input variables as new features. This makes it useful when the data shows a curved trend that a straight line can't capture.



3.3 Random forest Regression

Random forest regression is an ensemble learning method that builds multiple decision trees and combines their outputs to make more accurate and stable predictions. Each tree is trained on a random subset of the data, and the final prediction is usually the average of all tree outputs. It handles complex, nonlinear relationships well and is robust to overfitting and outliers.

4 CLASSIFICATION ALGORITHMS

Choosing the right classification algorithm is very important. An algorithm that performs classification is called a classifier. A classifier algorithm should be fast, accurate, and sometimes, minimize the amount of training data that it needs. Generally, the more parameters a set of data has the larger the training set for an algorithm must be. Different classification algorithms have different ways of learning patterns from examples. [1] More formally, classification algorithms map an observation v to a concept/class/label w.

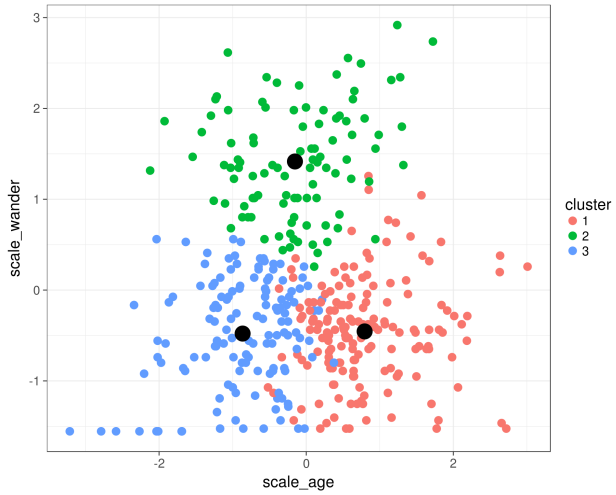
Many times, classification algorithms will take in data in the form of a feature vector which is basically a vector containing numeric descriptions of various features related to each data object. For example, if the algorithm deals with sorting images of animals into various classes (based on what type of animal they are), the feature vector might include information about the pixels, colors in the image, etc.

4.1 K-Means

The K-Means algorithm is an unsupervised machine learning technique used for clustering data into distinct groups based on similarity. The main goal of K-Means is to partition a dataset into k clusters, where each data point belongs to the cluster with the nearest mean (centroid). The algorithm starts by randomly selecting k initial centroids, then iteratively assigns each data point to the nearest centroid and recalculates the centroids based on the new cluster memberships. This process continues until the assignments no longer change significantly or a maximum number of iterations is reached.

K-Means is widely used for tasks such as customer segmentation, image compression, and pattern recognition because of its simplicity and efficiency, especially on large datasets. However, it has some limitations: it requires the number of clusters (k) to be specified beforehand, and it can be sensitive to

the initial placement of centroids and to outliers. Despite these challenges, it remains a foundational algorithm in unsupervised learning, often used as a baseline or as a component in more complex models.



4.2 Perceptrons

A perceptron is an algorithm used to produce a binary classifier. That is, the algorithm takes binary classified input data, along with their classification and outputs a line that attempts to separate data of one class from data of the other: data points on one side of the line are of one class and data points on the other side are of the other. Binary classified data is data where the label is one thing or another, like *yes* or *no*; 1 or 0; etc.

The perceptron algorithm returns values of $w_0, w_1, ..., w_k$ and b such that data points on one side of the line are of one class and data points on the other side are of the other. Mathematically, the values of w and b are used by the binary classifier in the following way. If $w \cdot x + b > 0$, the classifier returns 1; otherwise, it returns 0. Note that 1 represents membership of one class and 0 represents membership of the other. This can be seen more clearly with the AND operator, replicated below for convenience.

The perceptron algorithm is one of the most commonly used machine learning algorithms for binary classification. Some machine learning tasks that use the perceptron include determining gender, low vs high risk for diseases, and virus detection.

4.3 Naive Bayes Classifier

Naive Bayes classifiers are probabilistic classifiers with strong independence assumptions between features. Unlike many other classifiers which assume that, for a given class, there will be some correlation between features, naive Bayes explicitly models the features as conditionally independent given the class.

Because of the independence assumption, naive Bayes classifiers are highly scalable and can quickly learn to use high dimensional (many parameters) features with limited training data. This is useful for many real world datasets where the amount of data is small in comparison with the number of features for each individual piece of data, such as speech, text, and image data.

4.4 Decision Trees

Another way to do a classification is to use a decision tree. Say you have the following training data set of basketball players that includes information about what color jersey they have, which position they play, and whether or not they are injured. The training set is labeled according to whether or not a player will be able to play for Team A.

Person	Jersey Color	Offense or Defense	Injured?	Will they play for Team A?
John	Blue	Offense	No	Yes
Steve	Red	Offense	No	No
Sarah	Blue	Defense	No	Yes
Rachel	Blue	Offense	Yes	No
Richard	Red	Defense	No	No
Alex	Red	Defense	Yes	No
Lauren	Blue	Offense	No	Yes
Carol	Blue	Defense	No	Yes

What is the rule for whether or not a player may play for Team A? **They must have a blue jersey and not be injured.**

To use a decision tree to classify this data, select a rule to start the tree. We used "jersey color" as the root node. Next, we will include a node that will distinguish between injured and uninjured players, and we will reach our conclusion.

4.5 Support Vector Machines

Support vector machines are a supervised learning method used to perform binary classification on data. They are motivated by the principle of optimal separation, the idea that a good classifier finds the largest gap possible between data points of different classes.

For example, an algorithm learning to separate the United States from Europe on a map could correctly learn a boundary 100 miles off the eastern shore of the United States, but a much better boundary would be the one running down the middle of the Atlantic Ocean. Intuitively, this is because the latter boundary maximizes the distance to both the United States and Europe.

The distance from the SVM's classification boundary to the nearest data point is known as the margin. The data points from each class that lie closest to the classification boundary are known as support vectors. If an SVM is given a data point closer to the classification boundary than the support vectors, the SVM declares that data point to be too close for accurate classification. This defines a 'no-man's land' for all points within the margin of the classification boundary. Since the support vectors are the data points closest to this 'no-man's land' without being in it, intuitively they are also the points most likely to be misclassified.

4.6 Logistic regression

Logistic Regression is a statistical approach and a Machine Learning algorithm that is used for classification problems and is based on the concept of probability. It is used when the dependent variable (target) is categorical. It is widely used when the classification problem at hand is binary; true or false, yes or no, etc. For example, it can be used to predict whether an email is spam (1) or not (0).

Logistics regression uses the sigmoid function to return the probability of a label. Sigmoid Function is a mathematical function used to map the predicted values to probabilities. The function has the ability to map any real value into another value within a range of 0 and 1.

The rule is that the value of the logistic regression must be between 0 and 1. Due to the limitations of it not being able to go beyond the value 1, on a graph it forms a curve in the form of an "S". This is an easy way to identify the Sigmoid function or the logistic function. In regards to Logistic Regression, the concept used is the threshold value. The threshold values help to define the probability of either 0 or 1. For example, values above the threshold value tend to 1, and a value below the threshold value tends to 0.

4.6.1 Type of Logistic Regression

- **Binomial**: This means that there can be only two possible types of the dependent variables, such as 0 or 1, Yes or No, etc.

- **Multinomial**: This means that there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- **Ordinal**: This means that there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

Linear Regression is similar to Logistic Regression but different in some key implementation details, but more importantly in the interpretation of the results. While Linear Regression assumes that there is a linear relationship between dependent and independent variables. It uses the line of best fit that describes two or more variables. Its aim is to accurately predict the output for the continuous dependent variable. However, Logistic regression predicts the probability of an event or class that is dependent on other factors, therefore the output of Logistic Regression always lies between 0 and 1.

5 THE TASK

As a team, your job is to take on a dataset (of your own choosing) and integrate all your knowledge of data analysis to produce a coherent compelling story for your data. This story will have to conclude with something new that you learned from the raw dataset.

5.1 Required elements

To give you a roadmap on what would be expected out of your project, these are the topics that you should cover.

- **Data set**: Choose a data set that is compelling to you and your team and that is simple enough to extract some basic understanding quickly.
- **Data description**: You should include a general data dictionay of your data that should include some basic statistical values according to its domain. Things like range, mean value, std. deviation, etc.
- **Cleaning process**: Your data should be error free and ready to be processed by your algorithm. You should describe what where the issues you had to deal with (missing data, wrong data, wrong format, etc.) and what were the decision taken to solve them with a clear rationale for the conclusion.
- **Data visualization**: Some visual representation of the raw data should be part of your delivery.
- **Modeling/Feature Engeneering**: You should document what were the steps taken to encode your data and how did you decide what to use/discard for your project based on your research goal.
- **Storytelling**: Since the goal of the project is for you to discover something in the dataset, you should guide the reader with a compelling story on how you found what you found, and why should the reader care about it. To do this, I suggest you use also data visualization tools, such that it becomes clear what were your discoveries.
- **ML work**: You should state what ML algorithm you decided to use with a clear rationale. This should be consistent with the challenges you described earlier on the cleaning process, feature engeneering and the outcomes found.

5.2 How is it going to be graded?

Given that you are getting the creativity freedom to tell your story as you want, there is not a specific thing that should be shown in the document. Nevertheless, the maximum grade you can reach will be defined by the level of detail you produce.

The break of each of the grade will be:

- **10%** Report Quality
- **20%** Data preparation
- **30%** Descriptive Storytelling
- **30%** Algorithm Implementation
- **10%** Technical Achievement

5.3 What to deliver

- Source code of the ML implementation ready to execute with clear instructions on how to install, configure and run.
- A document specifying the analysis and details as described earlier.
- Development log. This is just a document with the explanation of the challenges you faced and how you solved them. It is perfectly OK to state that some of the challenges were not solved. If you refer to other sources to solve a problem, please don't forget to cite them properly. **The delivery of this document IS mandatory. It will not gives no credit, but the lack of it can deduct 5% of your grade.**

The code will be delivered via github