Politecnico di Torino

Master's Degree in Biomedical Engineering
A.Y. 2024/2025

Formulation, Characterization, and Machine Learning Prediction of Poly(Lactic-co-Glycolic) Acid Nanoparticles for Oncological Pregnant Women Treatment.

Supervisors:

Prof. VALENTINA ALICE CAUDA
Prof. CRISTINA FORNAGUERA
Prof. MARTA HERNÁNDEZ

Candidate:

VINCENZO GLORIOSO

Abstract

Cancer treatments during pregnancy pose serious risks, particularly chemotherapy, due to the potential transplacental passage of drugs that may attack the foetus. Such exposure can lead to malformations or, in severe cases, miscarriage. Despite these concerns, there is a clinical need for effective and safe cancer therapy for pregnant patients.

The objective of this thesis is to develop a nanoparticle-based drug delivery system capable of reducing foetal exposure to chemotherapeutic agents, while simultaneously introducing a novel machine learning driven strategy to optimize the nanoparticle design process. Specifically, polymeric nanoparticles were synthesized using Poly-(Lactic-co-Glycolic) Acid to encapsulate doxorubicin and minimize its transplacental transfer. Nanoparticles were prepared via a double emulsion method (water-in-oil-in-water), testing over 50 formulations by varying the proportions of aqueous phase, oil phase, and surfactant. Each formulation was systematically evaluated in terms of particle size, polydispersity index, zeta potential, encapsulation efficiency and drug loading.

A key novelty of this work lies in the integration of machine learning techniques to guide and accelerate formulation development. Supervised Machine Learning algorithms were trained to predict both the likelihood of stable nanoemulsion formation and the physicochemical properties of the resulting nanoparticles. This approach not only reduced experimental workload and material waste but also achieved predictive accuracies above 80%, demonstrating its potential as a powerful tool for formulation design.

By combining experimental nanoparticle synthesis with computational machine learning modeling, this thesis provides a dual strategy for creating a drug delivery systems safer and more efficient than current solutions for use in pregnancy; addressing both the medical challenge of maternal cancer treatment and the methodological innovation of applying Machine Learning to nanomedicine.

Table of contents

List of figures

List of tables

1 Motivation and Objectives

1.1 Motivation

Recent studies show that breast cancer affects approximately 1 in 10,000 women, this figure can transform and go from 1 in 3,000 if we consider a pregnant woman under the age of 45. [1] According Ruiz et al. (October 2017) “the rate of pregnancy associated breast cancer (PABC) varies from 2.6% to 6.9% of cases, which is almost 25000 new cases per year in the world”.

As extensive epidemiological research focused on the link between pregnancy and breast cancer risk, it became clear that the long-term protective effect of pregnancy is not uniform, but rather depends on the age at which a woman experiences her first pregnancy. [2]

Thinking about the impact of breast cancer on a pregnant woman, this concerns not only the condition of the mother and foetus, but also the psychological aspect, considering all the risks and comorbidities that can be faced. Some treatments, for example, can lead to the loss of the foetus or, if it is decided to continue the pregnancy, the patient may experience a worsening of her condition, up to, in the most serious cases, death.

Taking all this into consideration, the only treatment that does not pose significant risks to the foetus is surgery. Although it is an invasive procedure, it is generally considered the most suitable in all stages of pregnancy. Another treatment for breast cancer is radiation therapy, which, unlike surgery, is risky during pregnancy and is usually indicated only after giving birth.

Chemotherapy can be administered during the second trimester and after delivery, but it requires careful consideration of drug selection and dosage to minimize transplacental passage and reduce the risk of miscarriage or foetal malformations. Only a limited number of agents are considered relatively safe during pregnancy, including doxorubicin, cyclophosphamide, and paclitaxel.

In this case, doxorubicin is chosen due to its established efficacy in breast cancer treatment. Nevertheless, its administration is associated with potential adverse effects, such as cardiotoxicity, myelosuppression, alopecia, and mucositis, which necessitate vigilant monitoring throughout therapy.

In this regard, we are trying to develop an innovative approach through the use of carriers based on targeted drug delivery systems. These carriers are made up of the already approved PLGA

(polylactic-co-glycolic acid) nanoparticles, obtained using double water-in-oil-in-water nanoemulsion techniques.[3]

Among the main advantages of this method is the possibility of obtaining selective targeting in the area affected by the tumor, minimizing the risk of drug release in unwanted areas.

The use of PLGA therefore represents a promising strategy to deliver anticancer drugs in a targeted and safe way, and it is particularly relevant that this polymer has already been approved by the FDA for use in clinical settings.[3]

1.2 Objectives

This Master Thesis is part of a broader research project carried out in our group, which aims to design safe and effective drug delivery systems for cancer therapy during pregnancy. The overarching ambition of the project is to develop polymeric nanoparticle formulations capable of encapsulating chemotherapeutic drugs, reducing their transplacental transfer and thereby protecting the foetus while maintaining therapeutic efficacy for the mother.

Within this framework, **the specific objective of my thesis** is to investigate **the formulation of PLGA-based nanoemulsions and nanoparticles, and to explore how machine learning can accelerate and optimize their design**.

On the experimental side, my work focus is:

- Studying the effect of varying water, oil, and surfactant concentrations on the formation of nanoemulsions and nanoparticles, with the aim of constructing a ternary phase diagram that maps the domains where nanoemulsion formation is possible. These formulation studies are coupled with a detailed characterization of the resulting nanoparticles, assessing their size, polydispersity index (PDI), zeta potential, encapsulation efficiency, and drug loading capacity.

The computational part of this thesis introduces machine learning as a novel tool in nanoparticle formulation, and its focus is:

- Training a supervised machine learning model to predict the formation of stable nanoemulsions for any given composition within the ternary diagram. The predictive performance of the model is subsequently validated against experimental data, ensuring its reliability.

- Designing a second machine learning framework, building upon this foundation, to estimate key physicochemical properties of nanoparticles (including size, polydispersity index, surface charge, and encapsulation efficiency) directly from formulation variables. This computational strategy significantly reduces the reliance on extensive empirical testing.

By combining these experimental and computational approaches, the ultimate goal of the thesis:

- Identifying the most promising PLGA nanoparticle formulation, which will serve as the basis for future studies involving the encapsulation of doxorubicin for potential application in pregnancy-associated breast cancer therapy.

2 Theoretical Background

2.1 Pregnancy Associated Breast Cancer (PABC)

Pregnancy-associated breast cancer (PABC) refers to a delicate clinical condition that requires special attention. This form of breast cancer is diagnosed during pregnancy or within 1 year after birth. According to Paris et al. “The rate of PABC among pregnant women aged < 45 years has varied from 2.6% to 6.9%. In contrast, in those aged < 35 years, the rate has been 15.6% of all BC cases” [4]. Women who experience their first pregnancy after the age of 35 face an increased risk of developing PABC, and this elevated risk may persist for up to five years postpartum. With the current trend of delaying childbirth, it is likely that the overall number of breast cancer cases, including PABC, will rise in the future [4]. Since PABC typically occurs before the age of 40, there is an higher representation of breast cancer gene mutation carriers within this population [4-6].

The particularity of this pathology is that it does not only involve the health of the woman, but also that of the foetus. This makes diagnosis and treatment particularly complex: on the one hand, there is the need to intervene promptly to deal with cancer; on the other hand, you must be extremely careful not to put your pregnancy at risk. It is a challenge that requires a multidisciplinary approach, with oncologists, gynecologists, radiologists and pediatricians working together to ensure the best possible balance between effective care and foetal safety [1, 3-4].

During pregnancy, the breast undergoes numerous physiological changes (ducts branch out, lobes enlarge, blood flow increases and breast consistency changes); these changes are completely normal, but they may mask the presence of nodules or make them more difficult to detect by simple palpation. This is one of the reasons why PABC can be diagnosed late[7]. The most common symptom is a hard, fixed, painless lump that the woman notices on her own. However, because the breast naturally tends to change during pregnancy, many women (and sometimes even doctors) tend to underestimate or attribute these signs to hormonal changes [4]. From a biological point of view, the tumors associated with pregnancy tend to be more aggressive than those of non-pregnant women: they are often of invasive ductal type, with high cell proliferation (high Ki-67), and a higher frequency of negative and HER2-positive triple subtypes. This explains why many diagnoses occur in advanced stages, making treatment more urgent and delicate [1, 4-5].

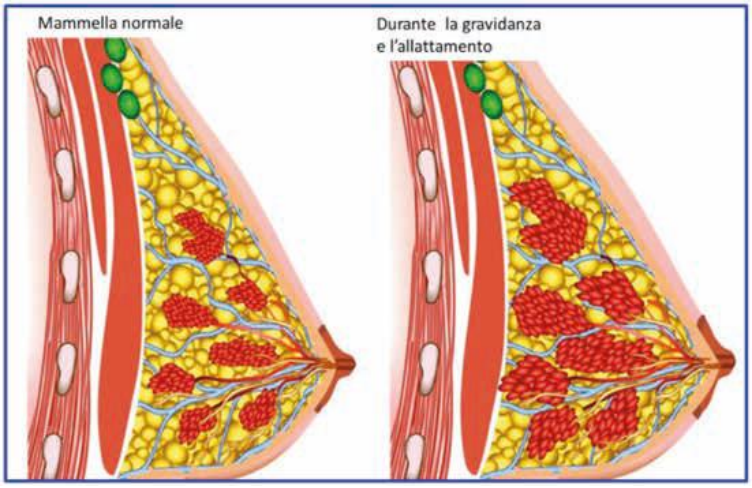


Figure 1 Difference of breast (left side: normal women; right side: during pregnancy) [8]

Although the general fear, it can safely perform tests during pregnancy in a safe way. The first check is a physical way where the attention is focus on the breast and the lymph nodes region. In addition, there is breast ultrasound, as it is safe for the foetus and very useful to identify suspicious masses cause of his sensitivity of 100% [4]. In the presence of a suspicious lesion, it is important not to delay the biopsy, which is the only way to get a certain diagnosis. It is crucial to remember that, with the right precautions, diagnostic tools do not pose a risk for the child, while a delay in diagnosis can worsen the prognosis of the mother. [1, 6-8]

Mammography is useful during pregnancy and lactation to assess disease extent, detect microcalcifications, and evaluate the opposite breast. Though less sensitive than ultrasound, it remains valuable. Foetal radiation exposure is extremely low and further reduced with shielding, making the procedure generally safe. [1, 6-8]

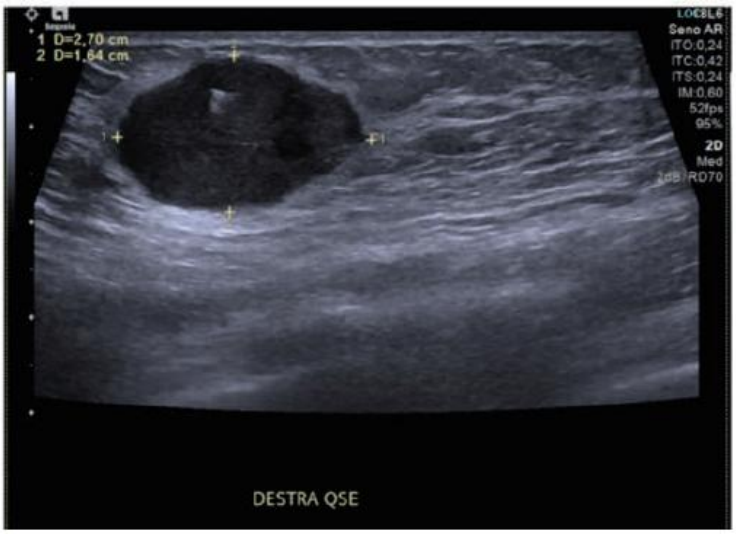


Figure 2 Breast ultrasound and detection of neoplasm [8]

The treatment of PABC should be decided by a multidisciplinary team and customized according to the trimester of pregnancy, the stage of the tumor and the patient's wishes. The surgery can be performed in any trimester; in some cases a mastectomy is chosen, in other cases a conservative surgery (with possible radiotherapy postponed to the post-partum). [1, 4, 8]

During the first trimester of pregnancy, chemotherapy is contraindicated because this amount of drugs can cause a risk of malformation of the foetus. Once you reach the second trimester, chemotherapy treatments can be used if necessary but considering the use of anthracyclines considered safe for certain circumstances such as pregnancy. In addition, when entering the third trimester of pregnancy, treatment is again not recommended to avoid complications later during childbirth. [3-4, 7, 9]

Radiotherapy, hormonal therapies, and targeted treatments are generally avoided during pregnancy due to their potential toxicity to the developing foetus. Radiotherapy is always postponed until after delivery because exposure to ionizing radiation, particularly during the first and second trimesters, can lead to serious foetal harm, including miscarriage, congenital malformations, and neurological damage [3, 5]. Hormonal therapies, such as tamoxifen, are contraindicated during pregnancy due to the high risk of birth defects and disruption of the hormonal balance essential for foetal development [9].

Similarly, targeted therapies, such as HER2 inhibitors like trastuzumab, are not administered during pregnancy because of potential complications including oligohydramnios and foetal renal toxicity [8]. The goal is not to compromise the effectiveness of cancer treatment, but rather to protect foetal development by avoiding therapies known to pose significant risks, especially during critical stages of gestation [3-4]. Although several therapeutic strategies have been established for the management of pregnancy-associated breast cancer, this work focuses specifically on nanoparticle-based drug delivery systems, highlighting their potential to enhance treatment efficacy and minimize foetal risk.

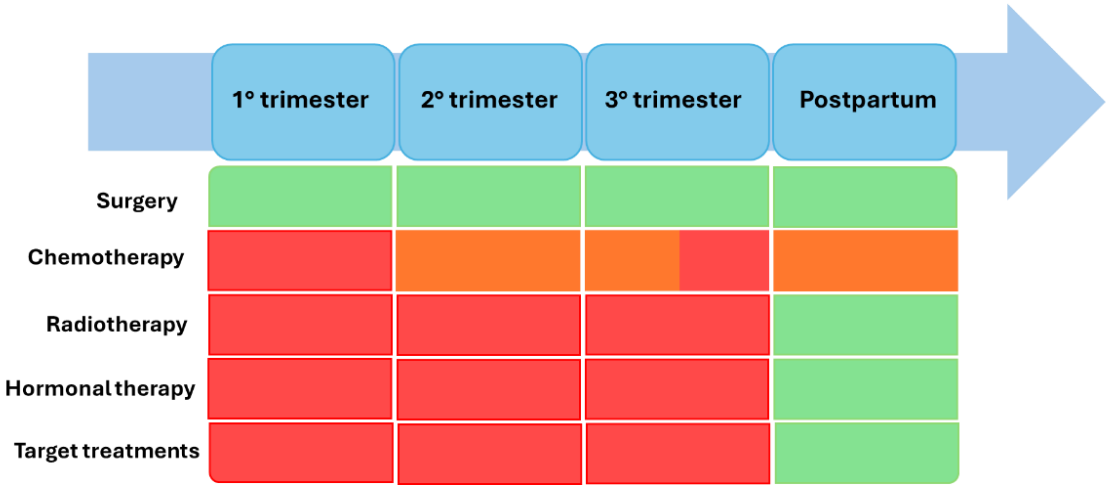


Figure 3 Timeline treatment during trimester (green: indicated treatment; red: contraindicated treatment; orange: caution treatment)

Discussing the prognosis of breast cancer during pregnancy may appear challenging; however, the availability of clinical data and knowledge in this area is significantly greater today than in the past. In general, if the cancer is diagnosed on time and treated properly, the prognosis of a woman with PABC may be similar to that of a non-pregnant woman with the same type and stage of cancer [7].

Another important aspect is the time when the tumor is discovered: some studies have observed that, when the tumor is diagnosed after delivery, the prognosis may be a little more unfavorable than a diagnosis made during pregnancy [12]. This may be due to changes in breast tissue during postpartum involution, a complex process that can make the microenvironment more "active" from a tumor point of view. Despite advances in diagnosis and treatment, these challenges highlight the urgent need for safer, more targeted therapeutic strategies that protect both mother and foetus. [1, 6-10]

The biological characteristics of the tumor also matter: if it is a more aggressive type, such as triple negative or HER2-positive, the disease may be faster in progression. Despite all this, the termination of pregnancy does not improve the mother’s survival and is not recommended, except in particular situations with very advanced disease and difficulty in treatment. [1, 6-8, 10]

To find a safer solution, further research has focused on optimizing nanoparticle design to enhance specificity and minimize off-target effects. This opens the door to exploring new strategies that go beyond traditional delivery systems.

2.2 Nanoparticles for Drug Delivery System (DDS)

Drug Delivery Systems (DDS) are carriers used in nanomedicine to transport therapeutic agents directly to a specific target in the body. This system operates at the nanoscale, using carriers typically smaller than 500 nanometers. In nanomedicine, drugs are incorporated into nano-sized carriers, such as nanoparticles, that can improve the precision and effectiveness of treatment. These nanoparticles can also have multiple functions: besides delivering the drug, they may be used for diagnostic imaging. This combination of therapy and diagnostics is known as theranostics. [11-15]

The use of nanoparticles in medicine, especially to bring drugs directly where they are needed, offers several really interesting benefits.

First, they allow to increase the precision of the treatment, minimizing side effects because the drug is not unnecessarily dispersed throughout the body. They can also protect the active ingredient until the right time of release, thus improving the effectiveness of therapy [19]. Another advantage is that some nanoparticles can overcome biological barriers (such as the blood-brain barrier), which normally block many drugs. They can also be designed to recognise specific cells, such as cancer cells, thanks to the possibility of "customising" them [20].

On the other hand, there are also critical aspects to be considered. One of the main problems is that the long-term effects of these particles in the body are not yet well known: some could accumulate in organs such as the liver or spleen, causing problems which are still unpredictable. Large-scale production can also be costly and technically complicated, and not all nanoparticles are easy to make stable or safe [20]. Finally, there is still regulatory uncertainty because clear international guidelines for the safety assessment of these technologies are missing [16-18].

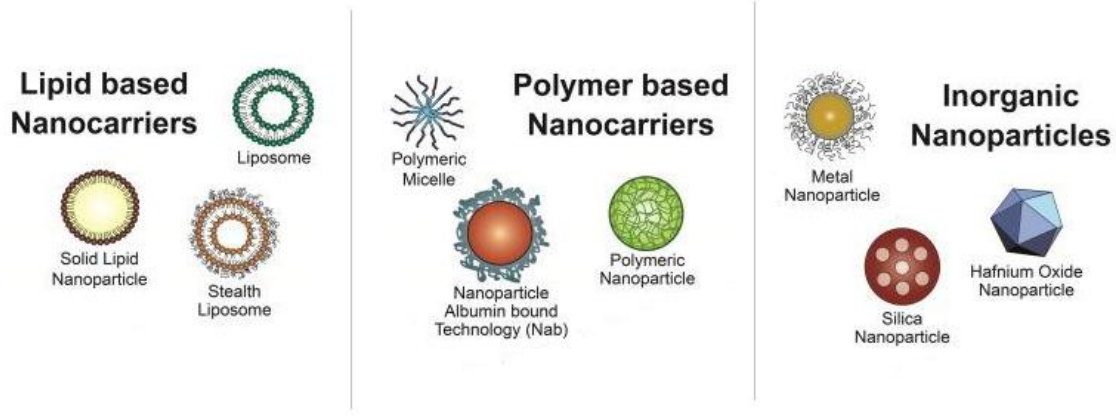


Figure 4 Different type of nanoparticles sort by materials [22]

The first big classification it can be according to materials: lipids nanoparticles, polymers nanoparticles and inorganics nanoparticle. Lipid nanoparticles are nanoscale carriers made from lipids, designed to mimic the structure of cell membranes. These systems, also known as lipid nanocarriers, can exist as either solid or hollow structures. When they have a hollow core, they are called liposomes. Liposomes can also be engineered to have stealth properties, allowing them to evade the immune system and circulate longer in the body. [11, 13-15]

Polymeric nanoparticles can be solid, capsule or nanocomplex, according to Beach et al. "A controllable size, shape, and surface charge are three of the principal advantages associated with polymeric nanoparticles for drug delivery." [15] Polymer nanoparticles represent an advanced frontier in targeted drug delivery, offering numerous benefits. They enable the targeted crossing of complex biological barriers, such as endothelial and intracellular barriers, to encapsulate a wide range of active ingredients, including small molecules, peptides and nucleic acids, to ensure a controlled release of the drug in response to specific stimuli, such as changes in pH, presence of enzymes, temperature or redox agents. [11-15]

Inorganic particles are obtained from non-polymer materials, such as metal particles. Typically, they are massive, but not necessarily because it is possible to obtain nanoshells of gold, hollow particles. If present the drug will be outside, they are usually not used for the release of drugs but to do imaging. The same nanoscale material may have some specific therapeutic effect, some can be used for thermal therapy of tumours. [11,13-15]

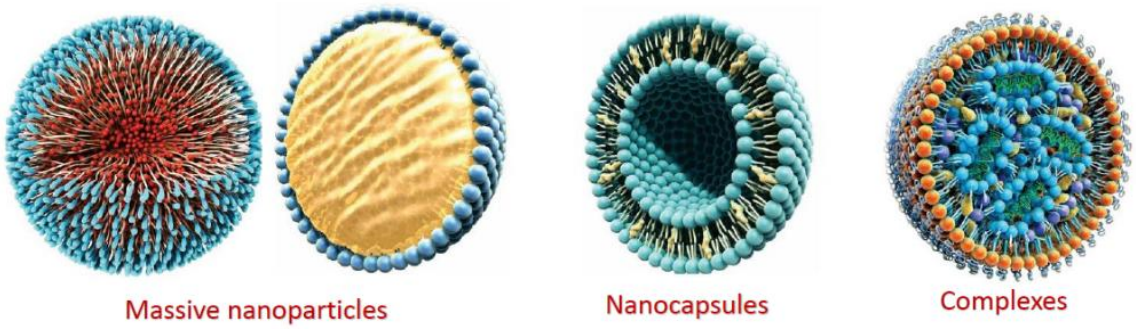


Figure 5 Different type of nanoparticles sort by physical structure [23]

We can make a second classification taking in consideration the physical structure: we distinguish solid nanoparticles if they have a full structure and the drug is dispersed within the matrix in a uniform way; we can have, instead, nanoparticles with hollow structures called nanocapsules in which the drug may be inside or outside the shell; finally we can have a nano

complex or nanoparticle that exploits the electrostatic interaction between the material of the nanoparticle and the drug.[15, 21].

Different types of targeting can be distinguished: active, passive and stimulus-mediated. Active targeting requires site-specific recognition, such as a binding protein to modify the surface of the particle with a ligand that is specific to the receptor on the cancer cell [25]. Passive targeting is typical only of small nanoparticles, which are able to selectively permeate through the fenestrations by accumulating in tumors [25]. In stimulus-mediated targeting, on the other hand, once the particles are at the site to activate they need an external stimulus which can be a change in PH, temperature or concentration of enzymes. These particles are sometimes also referred to as intelligent nanoparticles [14-16, 22].

2.3 Poly-(Lactic-co-Glycolic Acid) (PLGA)

One of the materials often used for drug delivery nanoparticle is PLGA, short for poly-(lactic-co-glycolic acid). It is a copolymer made from two basic building blocks: PLA (polylactic acid) and PGA (polyglycolic acid). What makes PLGA special is that it's biodegradable, meaning it breaks down naturally in the body thanks to water or enzymes, without leaving toxic residues. It's also approved by authorities like the FDA and EMA, which makes it a safe and reliable option for medical use. [23-27]

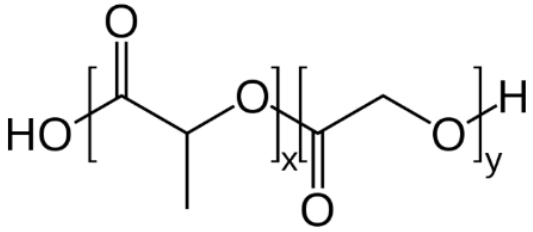


Figure 6 Structure of PLGA [29]

During its production, which involves a polymerization reaction, we can adjust how strong or fast degrading the material is just by changing the ratio between PLA and PGA. Another key point is its biocompatibility: the body doesn't see it as a threat, so it doesn't trigger inflammation or rejection. [23-27]

PLGA is flexible: it can be used to make scaffolds, thin films, or nanoparticles, depending on how we want the drug to be released and how quickly we want it to degrade. When used as a drug carrier, PLGA tends to release the active ingredient slowly over time, which is great for long-term treatments. That said, in some cases, there can be an initial burst release, where a big

dose comes out all at once, which isn't always ideal. Luckily, scientists are working on ways to fix this, for example, by designing particles that respond to things like pH or temperature, so the drug is only released in certain conditions. [23-27]

PLGA is also being widely tested in cancer treatment, especially to carry chemo drugs like doxorubicin and paclitaxel straight to the tumor. This helps make treatments more targeted and reduces damage to healthy cells. Still, there are a few challenges. Some studies have shown that the particles can build up in organs like the liver and spleen, which raises concerns about long-term use. Also, making these nanoparticles on a large scale is still quite tricky and expensive, which makes it harder to bring them into everyday clinical practice. [23-27]

2.4 Nanoparticle formulation

There are different types of nanoparticle formulation using preformed polymers like emulsion, nanoprecipitation and salting out. Nanoemulsions are suspensions of particles of one liquid in another liquid, we use the prefix nano to define the scale of dimensions (from 20 to 500 nm), the emulsion is formed if the two liquids are immiscible [30].

They are defined as suspensions because, when oil and water are mixed, the liquids remain separated unless stirred. In this case, the oily phase is forced to disperse finely in the water; however, if the emulsion is left to rest, the phases will separate again after a certain period of time [31]. To prolong the stability of emulsions, a stabilizing agent, also known as a surfactant, can be added. This is an amphiphilic molecule that positions itself at the interface between water and oil. [30].

While shaking, the system is destabilized and the emulsion is formed, the surfactant is deposited at the interface between the drops and slows down the tendency of the drops to collide and unite with each other, It makes them more stable and slows down the coalescence between drops and the emulsion will take longer to return to the initial condition. [17, 30-33]

This feature of emulsions is used to obtain polymer nanoparticles because the oil can be a solvent that dissolves the polymer in which you want to make the nanoparticles, stirring will form drops of oil in the water containing the polymer. If the solvent is evaporated, the droplets dry up and become suspended spheres that can be collected, they are polymer nanoparticles [30].

This is called oil emulsion in water (O/W), the oil is the phase present in smaller volume and, stirring, it will be the drops of oil to disperse in the water. It is possible also do the opposite, if

the water and the phase present in smaller quantities when you shake will be the water drops to disperse in the oil phase, this is an emulsion water in oil (W/ O) [31].

These emulsions exploit the single emulsion, it is stirred once and forms a single emulsion between the solvent and water, but you can also emulsify more than once: there are techniques of double emulsions, these improve the effectiveness of encapsulation of drugs [30]. To make a double emulsion it takes more than two phases, for example if we make an emulsion water in oil in water (W/O/W) you will have three phases: an internal aqueous phase, an oil solvent phase and a second aqueous phase called the external phase. [30-33]

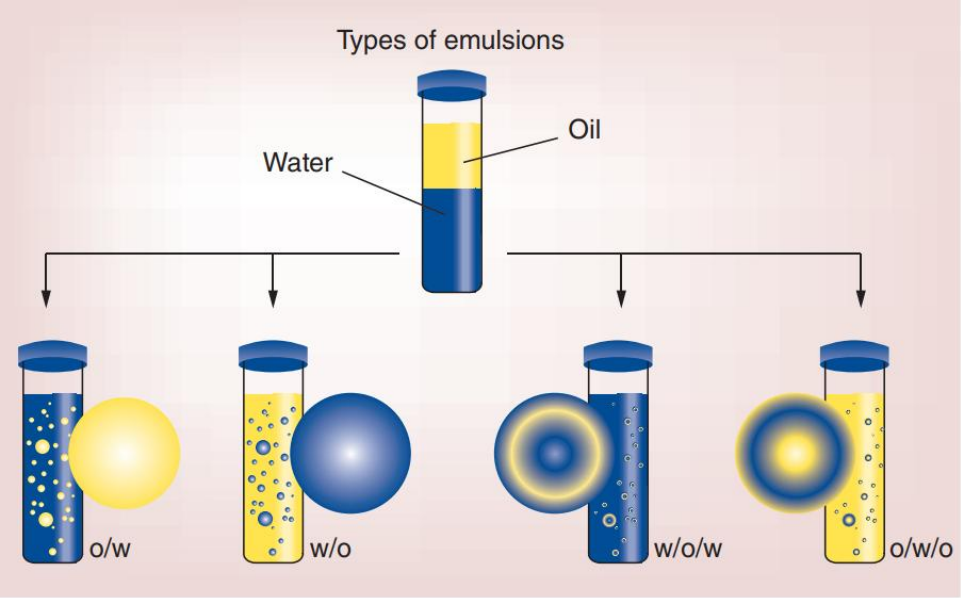


Figure 7 Type of nanoemulsion: oil in water (O/W); water in oil (W/O); water in oil in water (W/O/W); oil in water in oil (O/W/O) [30]

The process is similar to the single emulsion: initially, a first emulsion is prepared in which the aqueous phase is present in smaller quantity, by stirring the mixture, a water-in-oil (W/O) emulsion is obtained. This primary emulsion is then shaken and mixed with a larger quantity of aqueous phase to form a second emulsion, resulting in a water-in-oil-in-water (W/OW) system [30].

At this stage, the solvent is evaporated, the droplets dry, and polymer particles remain containing the hydrophilic drug. The double emulsion is a scalable process, and the particle size can be controlled by adjusting the parameters during the different steps of the procedure [30].

The basic parameters of a nanoemulsion include the measurement of nanoparticle size, expressed in nanometres, which indicates the diameter of the particles, and the polydispersity

index (PDI), a value between 0 and 1 that provides information about the presence of nanoparticle aggregates, higher values indicating greater heterogeneity [30-33]

To better understand and fine-tune the formulation of these nanostructures, ternary phase diagrams offer a useful way to visualize how different components interact and influence the final system.

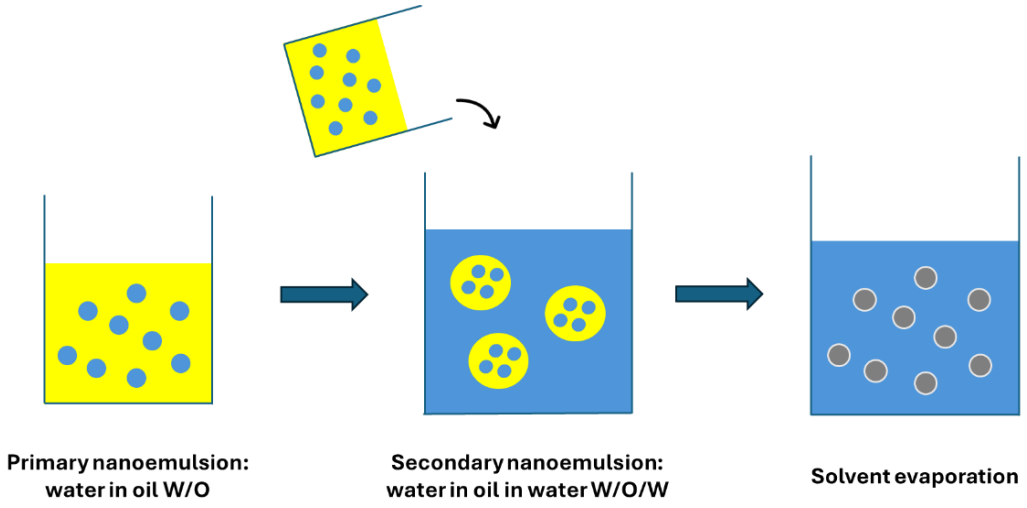


Figure 8 Step of double Nanoemulsions formulation water in oil in water W/O/W

2.5 Ternary diagram

A ternary diagram is a graph drawn within an equilateral triangle, used to represent the composition of systems formed by three components, whose sum is constant. This constant value is often expressed as a percentage and, for convenience, referred to as 100% [34]. The peculiarity of this type of diagram is that each vertex of the triangle represents the maximum purity of one of the three components, that is a condition in which only one substance is present at 100% while the other two are absent. Along the sides of the triangle are binary mixtures, that is combinations of two components, with the third absent. Finally, each point within the triangle represents a specific mixture of the three components in defined proportions. [34-37]

In order to facilitate reading and interpretation, the triangle is often divided into a grid of lines parallel to each side, which makes it possible to estimate relative percentages more accurately. A fundamental aspect of this type of representation is that, since the sum of the three components is constant, two values are enough to identify a point: the third is obtained automatically [38].

This implies that the compositions, although three-dimensional in theory, can be effectively represented in two dimensions, namely on the plane of the triangle. When working with ternary mixtures, the ternary diagram becomes a particularly effective tool because it allows all possible combinations of the three components to be displayed in one image [38].

This is very useful because, in most real cases, the quantities of substances are not independent of each other: if one increases, at least another must necessarily decrease. The diagram takes account of this interdependence, offering a coherent and realistic representation of the system. [34-37]

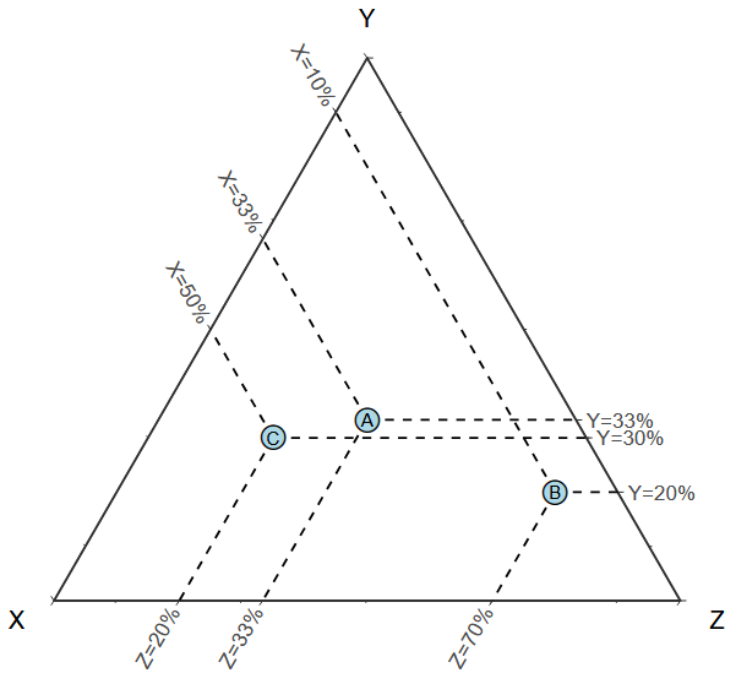


Figure 9 Different point in ternary plot [38]

From a practical point of view, this type of graph is valuable because it makes even complex systems visible and understandable which would otherwise be difficult to interpret with only numerical data. It also helps to identify non-immediate phenomena, such as phase transitions or abnormal behaviours that only emerge when the composition changes [36].

For this reason, it also allows more targeted prediction and design of experiments, based on the simple observation of areas in the graph showing the desired characteristics. Finally, the ternary diagram is considered a true common language between different disciplines such as chemistry, materials science or biology when it comes to interdependent three-component systems. [36, 37]

While ternary phase diagrams provide valuable insights into formulation behavior, the complexity of multivariable systems often requires more advanced tools to avoid hundreds of wet lab experiments, this is where machine learning can play a transformative role.

2.6 Machine Learning for nanoemulsion prediction

Artificial intelligence can be broadly categorized into three classes of techniques: expert systems, which are programs designed to solve problems in specific domains with performance comparable to that of a human expert; machine learning; and deep learning. Collectively, these approaches are referred to as intelligent systems, meaning computational systems capable of representing knowledge in a form that can be processed and interpreted by a computer according to a defined algorithm.

This knowledge can be set either externally by an expert or through an input dataset. According to S. Bini (2018) “Machine learning is best considered as a subset of AI. Machine learning learns from experience and improves its performance as it learns. As we saw in the earlier examples, it is a field which is showing promise in helping to optimize processes and resource allocation” [39].

We focus on machine learning because it enables us to make predictions through different learning models. This aspect becomes particularly relevant when considering its main advantages.

First, it allows a reduction in operating costs and waste, since computational models can estimate the desired characteristics in a specific context. Moreover, decision-making processes can be optimized by implementing algorithms based on historical data, employing both linear and nonlinear models to improve accuracy. Even with a limited dataset, machine learning can contribute to greater efficiency and productivity. However, alongside these advantages, certain limitations must also be taken into account.

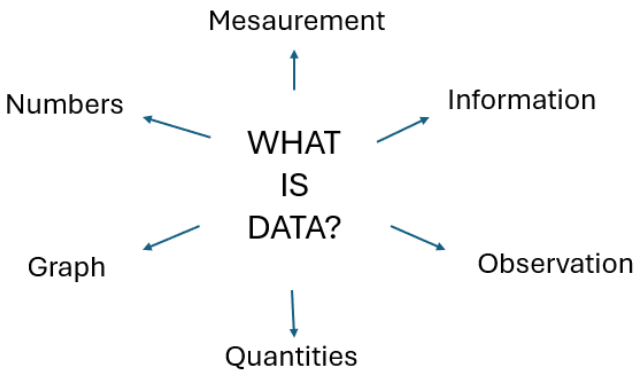


Figure 10 This photo questions different type of data mining

There are several methods for building a dataset. The first, and perhaps the most direct, consists in collecting data personally in the laboratory, through experiments conducted personally using the available machinery. This approach lends greater credibility to data, as it is collected directly and under control, and can lead to new and original results.

Alternatively, it is possible to build a dataset by drawing on data already present in published scientific articles. However, this method is more indirect: the quality and reliability of the data depend on the correctness of the procedures followed in the original studies, on the possible different calibration of the instruments used, or on possible undocumented malfunctions. Therefore, although useful and less time-consuming and resource-intensive, it has limitations from the point of view of data validity and traceability.

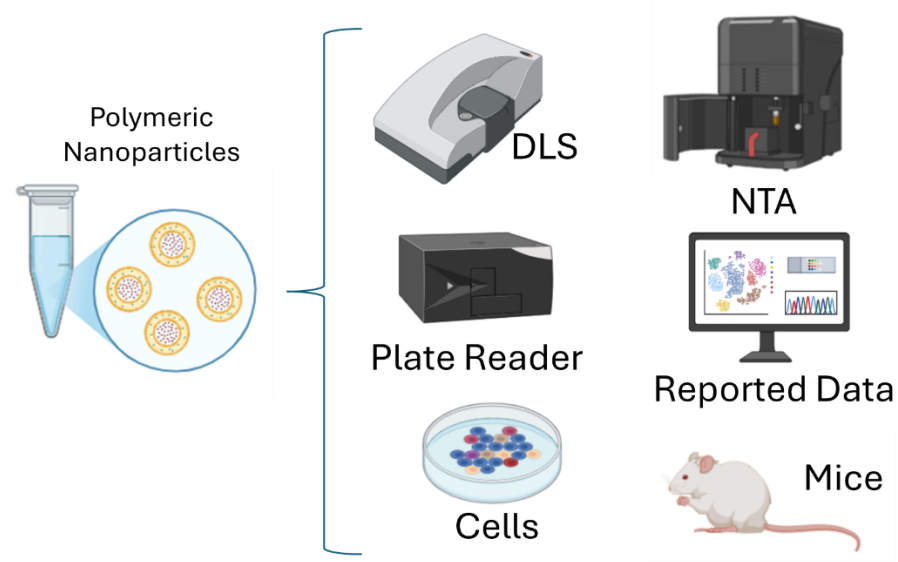


Figure 11 Different method of data mining to construct a dataset

The accuracy of the prediction heavily depends on the completeness of the input data; incomplete data can lead to unreliable predictions. Additionally, bias and discrimination may arise if the dataset is unbalanced, potentially causing the model to produce unfair results. Overfitting is another concern, as it can reduce the model's ability to generalize to new data effectively. Finally, machine learning models require continuous monitoring and periodic retraining to maintain their performance over time [37, 39-43].

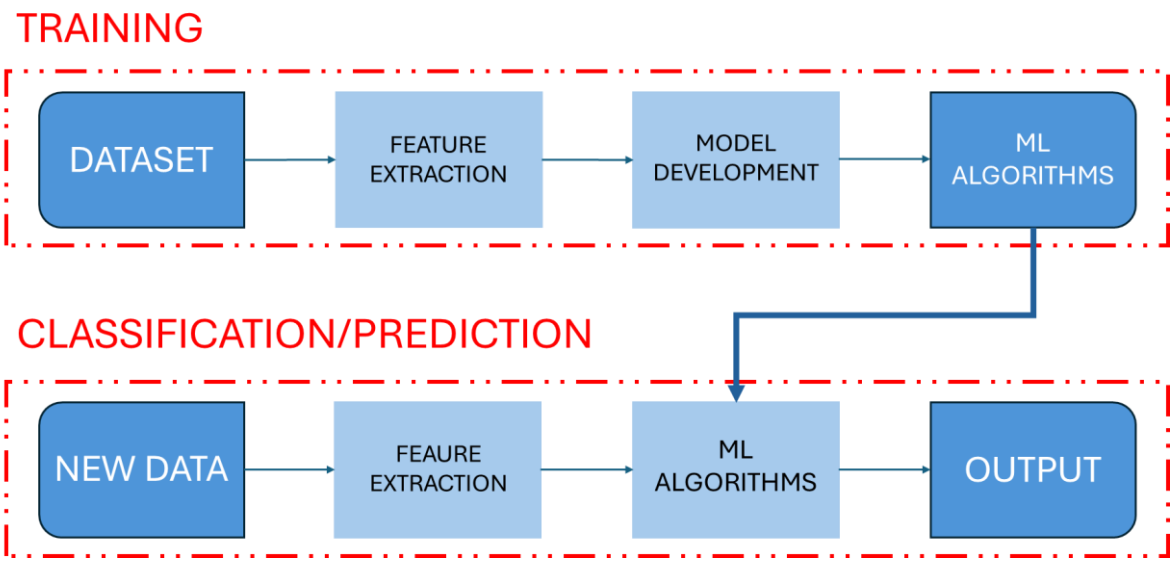


Figure 12 Process of training and classification/prediction of a machine learning algorithms

In general, there are two phases to a machine learning process. Initially a training phase, where starting from a dataset, called training set, we extract the variables that describe the data; these variables will be useful to build our algorithm. The second phase is the prediction phase, where giving new input variables to our algorithm returns a prediction in output. There are different types of learning which we can divide into three general types: supervised, unsupervised and reinforced learning. Supervised learning allows us to make predictions, that is by giving in input a series of independent variables (x) and in output a series of dependent variables (y) will model a relationship between the variables; Once you have the relationship when you are going to insert a new element x you can predict its corresponding element y. Unsupervised learning is used to divide training set into homogeneous groups so as to recognize patterns in data where you do not know the class. Reinforced learning, finally, is built through feedback that the algorithm receives from external stimuli [44-46].

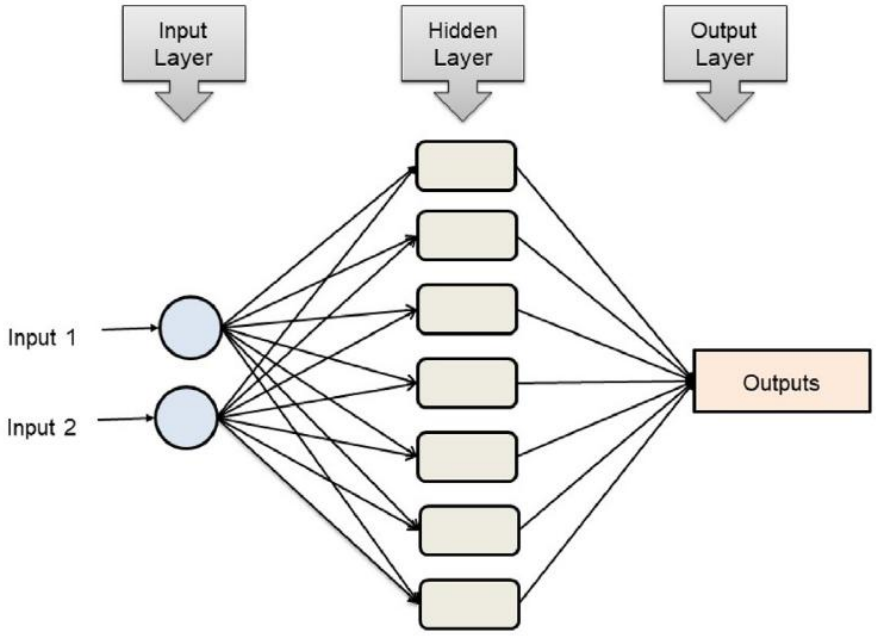


Figure 13 Symbolic configuration structure of the machine learning prediction [43]

In addition, in the various algorithms we can decide whether to have variables that do not affect each other so the model is based on a single variable to make predictions, this approach is called univariate, or we can have variables that influence each other going so we consider the result of the other variables going to capture more complex relationships, this approach is called multivariant.

In the field of supervised learning, several approaches can be employed depending on the nature of the data and the predictive task. Among the most widely used are algorithms such as K-Nearest Neighborhood (KNN), Support Vector Regression (SVR), Polynomial Regression, Random Forest and Gradient Boosting. Each of these methods is based on different principles and offers distinct advantages and limitations. In the following sections, their theoretical foundations and practical applications to the present work will be described in detail.

2.6.1K-Nearest Neighborhood (KNN)

The K-Nearest Neighborhood (KNN) is a very simple classifier, which is based on an intuitive method of going to look at nearby items. Given an element, it is classified based on classes and corresponds to the k-elements that have the smallest distances from the element itself.

It can be used both for prediction and regression. The advantages of using this model are the simple implementation, it is analytically tractable, it lends itself very easily to parallel implementations and uses local information in order to produce an adaptive behavior; the

disadvantages, on the other hand, are the large storage requirements and the computationally intensive recall. [46]

If the value of *K* is set too high, classification errors may occur, as elements that do not belong to the same class can be incorrectly grouped together. Conversely, if *K* is chosen to be too small, the model may fail to include data points that actually belong to the same class, thereby reducing classification accuracy. A common approach to obtain an initial estimate of *K* is to calculate the square root of *n*, where *n* represents the number of elements in the training set. [48-51].

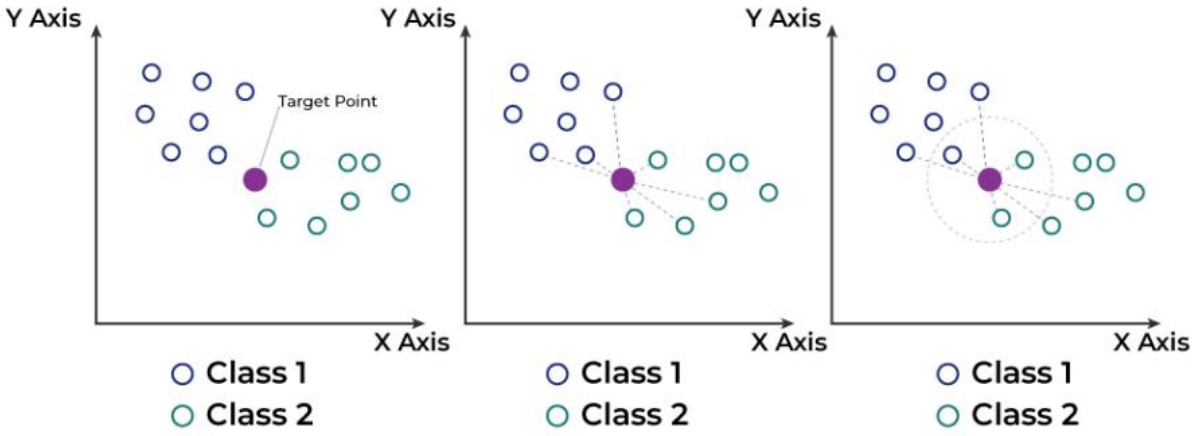


Figure 14 Time step of KNN classification/prediction method [52]

2.6.2 Support Vector Regression (SVR)

Support Vector Regression (SVR) is a machine learning methodology designed to solve regression problems, that is to predict numerical values rather than classes. Unlike traditional models that focus on minimizing the error between forecasts and actual values, SVR takes an alternative approach: it tries to identify a function that is as close as possible to the data, accepting small errors within a defined epsilon margin. [51]

In essence, if the error remains within this acceptable range, it is not considered a problem. Only the most significant errors are considered to make changes to the model. This makes the SVR particularly useful for avoiding overly rigid or data-sensitive models. [52]

Like the Support Vector Machines used for classification, it is also possible to use the so-called kernel trick, a technique that allows non-linear problems to be tackled by transforming data into more complex spaces, where it is easier to identify a valid regression function. This implies that the SVR can adapt also to data with non-linear relationships, while maintaining control over

the complexity of the model. The strengths of SVR include its ability to generalise effectively, flexibility in handling various types of data and good resistance to outliers. [51]

However, it can be quite computationally demanding, especially when working with very large data sets or many parameters to optimize. Nevertheless, it continues to be a good choice for those seeking a balance between precision and control, particularly in areas such as finance, bioinformatics or process engineering, where having both accurate and interpretable models is crucial [53-56].

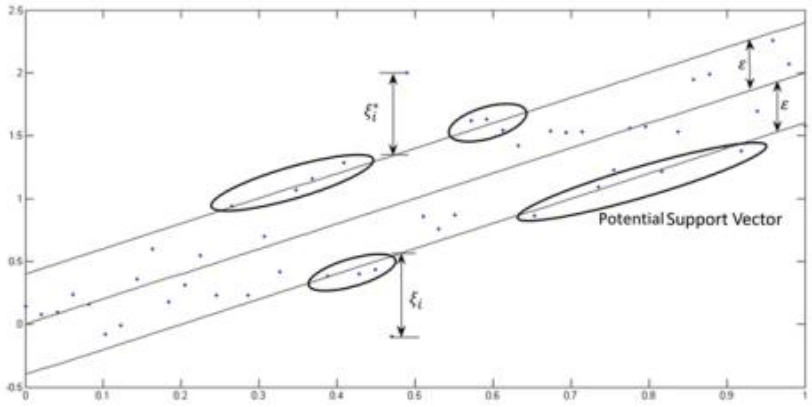


Figure 15 SVR classification/prediction method [57]

2.6.3 Gradient Boosting

Gradient Boosting is a machine learning technique capable of producing highly accurate models, especially in regression and classification problems. Its basic logic is simple but effective: instead of developing a single complex model, it creates a sequence of simpler models which are strategically combined.

Each successive model focuses on the errors made by previous ones, trying to correct them step by step; this iterative process allows you to progressively improve the performance of the system so that in the end, it is possible get a final model made up of many small corrections that allow high precision predictions. [56]

In the fundamental Gradient Boosting is its gradual approach to each iteration, which adjusts a parameter called learning rate; this parameter determines the extent of the correction made by the model and allows a controlled and better construction of the result so as not to have too drastic changes in the end. [57],

Although this technique is effective, some precautions must be taken: too many iterations can lead to an oversized model, suitable only for the data on which it has been trained but not

performing well on new datasets, can be mitigated by carefully adjusting the parameters and implementing strategies such as cross-validation or early stopping, which allows you to stop training at the optimal time. [56]

In conclusion, the Gradient Boosting is based on widely used algorithms such as XGBoost, LightGBM and CatBoost, its strength lies in the ability to improve progressively through learning from errors, managing to create solid models even in complex situations, provided that the right balance between accuracy and simplicity is maintained [58-60].

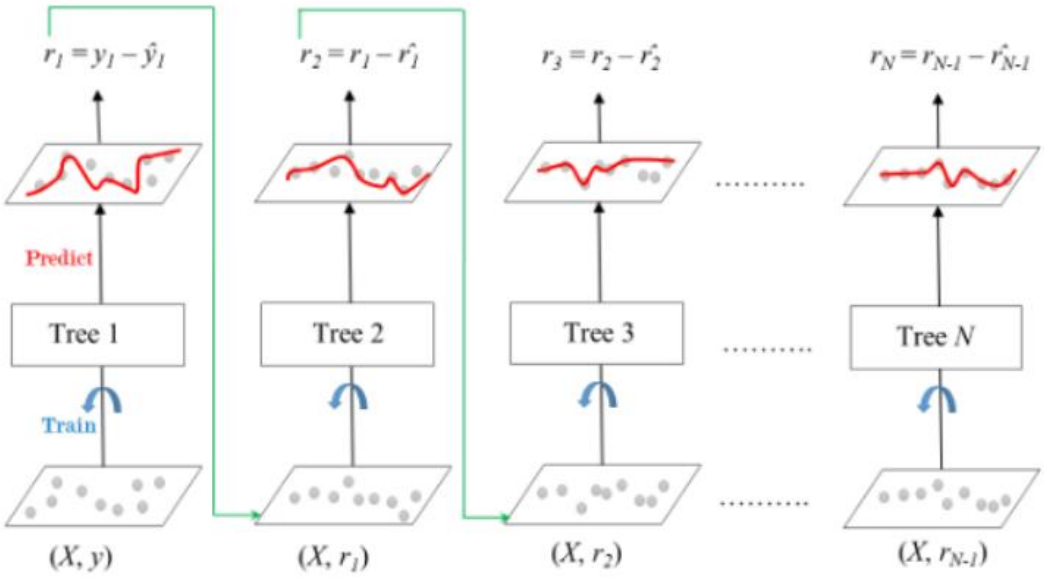


Figure 16 Gradient Boosted trees[61]

2.6.4 Polynomial regression

Polynomial regression is a variant of the classical linear model, used to describe more precisely situations where data show a non-linear relationship between variables; Instead of representing the data with a simple straight line, it creates a curve of greater degree than the first or more elaborate form, which better reflects the real behaviour of the data: incorporating higher powers allows the predictive function to adapt to more complex dynamics. [60]

A significant advantage of polynomial regression lies in its conceptual simplicity: while extending the classical linear approach, the model remains interpretable and manageable, able to represent non-linear phenomena with efficiency, but as for any modeling method, there are limits: an excessive increase in the degree of the polynomial can generate a curve that fits perfectly to the training data but loses reliability on new data, a phenomenon known as

overfitting; for this reason, it becomes essential to carefully select the degree of the polynomial, balancing flexibility and predictive capability. [60]

Polynomial regression is particularly useful when the relationship between variables shows non-linear variations, as in growth models, physical dynamics or long-term economic trends; The introduction of simple transformations of variables makes it possible to move from a straight line to a curve capable of capturing with greater precision the complexities present in the data. [62-64]

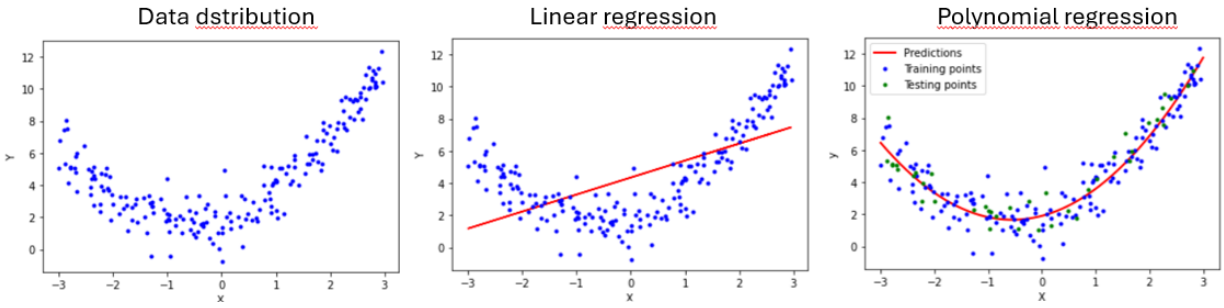


Figure 17 Difference between a linear regression and a polynomial regression [65]

2.6.5 Random Forest

The Random Forest algorithm is one of the most popular machine learning techniques for its ability to generate accurate and stable forecasts, even in the face of complex or disorganized data; It builds a number of slightly different trees and combines their responses to produce a more reliable final prediction; the idea is to rely on more experts with different opinions and find a common solution so a more balanced result. [64]

The way to have as many different decision trees as possible is to randomly choose the data to train each tree and select the variables on which each tree bases its decisions; in this mod the trees come to the same solution and it reduce the risk of overfitting. [64]

Random Forest offers many advantages, including excellent accuracy even in the presence of noisy or incomplete data; its versatility is remarkable: it is suitable for both classification and regression problems and, in addition, the algorithm suggests which factors have the greatest impact on the final results. [65]

On the other hand, there is a problem of interpretability: the presence of many trees makes it difficult to analyse in detail the process leading to a given forecast and the increase in the number of trees or the handling of large quantities of data can lead to higher computational

costs. The Random Forest stands out as a powerful, flexible and highly reliable algorithm, and is widely used in fields such as medicine, finance and marketing, where accuracy is crucial. [66-68]

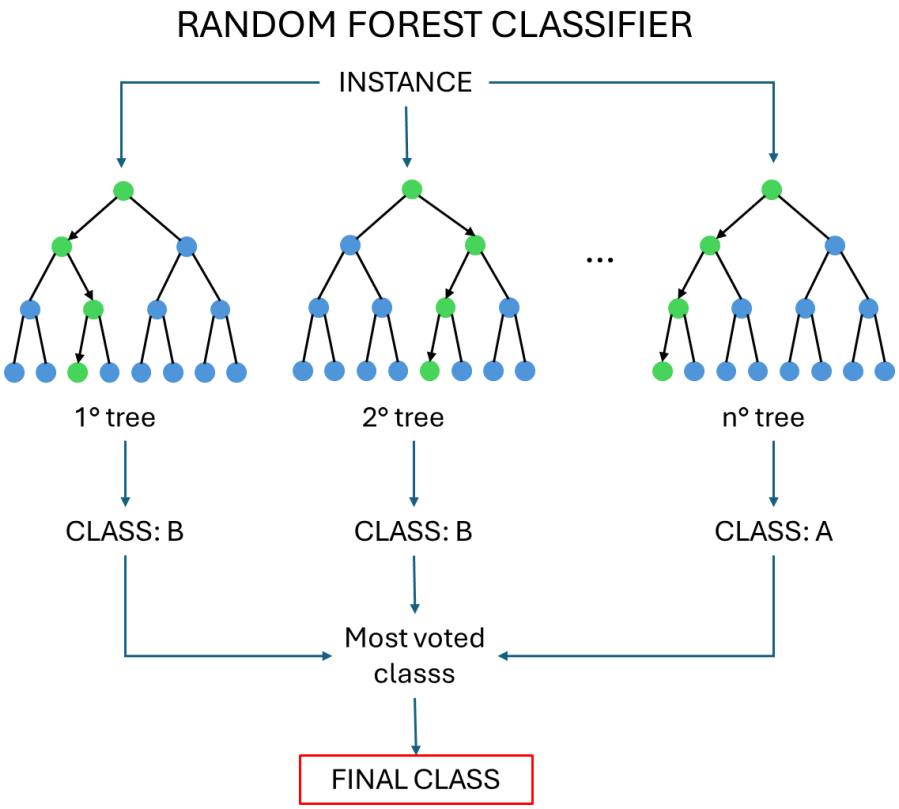


Figure 18 Random Forest classification/prediction method

2.6.6 K-Fold Cross-Validation

To validate the results before training the model you can separate from the dataset a quantity of data, which will be called validation set, used to monitor the progress of the network training.

When there is a limited amount of data, using a fixed validation set can greatly reduce the amount of information available for model training. To overcome this problem, k-fold cross-validation is often used, a technique that involves the subdivision of the dataset into k sub-sets (folds). [67]

The model is then trained k times, each time using a different fold as a test set, while the remaining k - 1 are used for training. The final error is calculated as an average of the errors obtained in k iterations. In addition, during each iteration, it is possible to reserve a portion of the training set for validation, useful for parameter optimization.

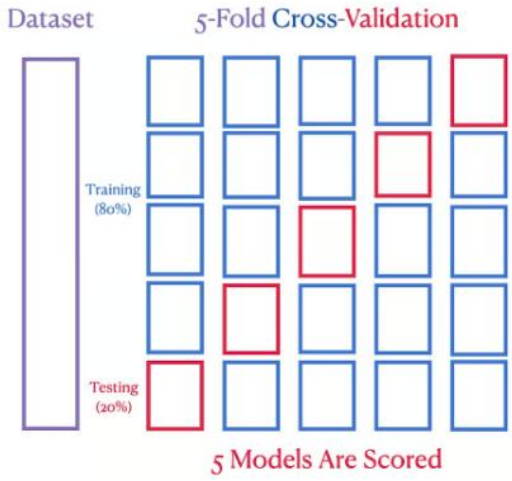


Figure 19 Cross validation scheme [69]

This approach allows to more reliably estimate model performance across the entire dataset and identify more robust parameter configurations. [70-73]

3 Materials and methods

3.1 Materials

Poly(D,L-lactide-co-glycolide) (PLGA) with a 50:50 molar ratio of lactic acid to glycolic acid and a molecular weight ranging from 24,000 to 38,000 Da was purchased from Sigma-Aldrich (Merck) (product code 719897, 5 g). Polysorbate 80 (Tween® 80), a non-ionic surfactant, was purchased from Sigma-Aldrich (Merck) (product code P4780, 500 mL). Pluronic® F-127 (poloxamer 407), a non-ionic triblock copolymer composed of poly(ethylene oxide) and poly(propylene oxide), was purchased from Sigma-Aldrich (Merck) (product code P2443, 250 g). Nile Red, a lipophilic fluorescent dye used for staining hydrophobic structures, was purchased from Sigma-Aldrich (Merck) (product code N3013, 100 mg). Rhodamine B, a fluorescent dye commonly used for imaging and tracer studies, was purchased from Sigma-Aldrich (Merck) (product code R6626, 25 g). Ethyl acetate (CAS number 141-78-6) was purchased from Sigma-Aldrich (Merck) (product code 23879, 5 L). Ultrapure water filtered and sterilized using the Synergy UV system was used throughout the experiments.

3.2 Methods

3.2.1 Nanoparticles preparation

Nanoparticles are prepared using a water-in-oil-in-water (W/O/W) double emulsion process.

First, a 1% (w/v) Pluronic F-68 solution in ethyl acetate is prepared. For the first emulsion (W/O), 10 mg of poly(D,L-lactide-co-glycolide) (PLGA) are placed in a 15 mL conical centrifuge tube (Falcon) together with 0.25 mL of Milli-Q water (10% v/v aqueous phase) and 2.25 mL of the Pluronic solution in ethyl acetate (90% v/v organic phase). The mixture is homogenized for 2 min at the maximum speed setting (6) of an Ultra-Turrax IKA T10 basic homogenizer.

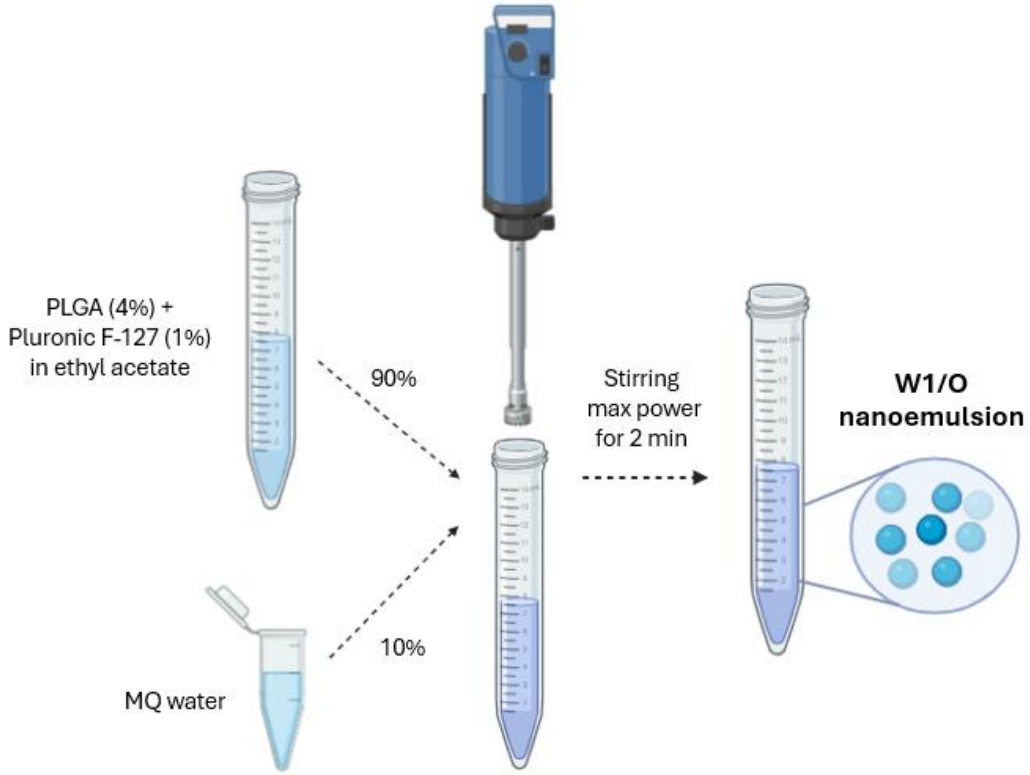


Figure 20 First nanoemulsion water in oil W1/O

For the second emulsion (W/O/W), the organic phase from the first emulsion is placed in a clean 15 mL Falcon tube together with the appropriate amount of Tween 80. The mass of Tween 80 is calculated from the required volume, based on the target surfactant percentage, using its density of 1.06 g/mL. The calculated volume of the organic phase from the first emulsion is then added. The mixture is blended using a Heidolph Reax Top vortex mixer at 3,200 rpm until the surfactant is completely dispersed in the organic phase.

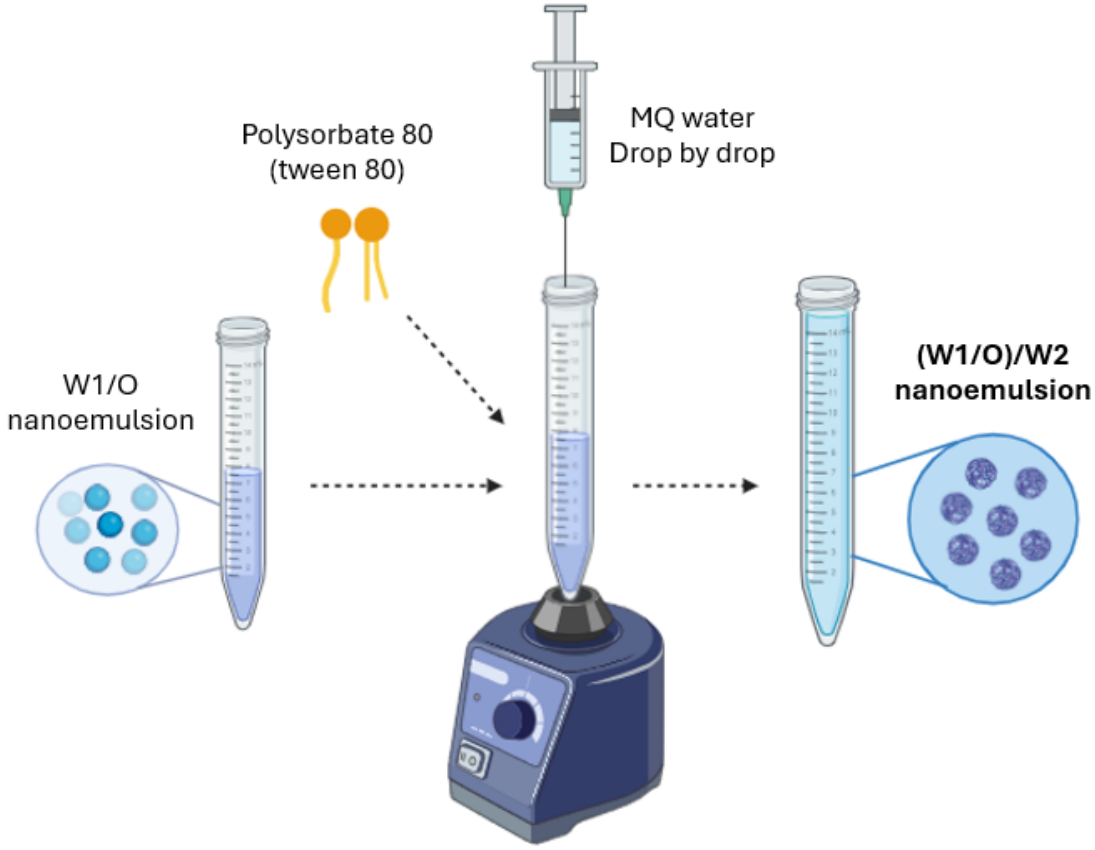


Figure 21 Second nanoemulsion Water in Oil in Water (W1/O)/W2

While keeping the Falcon tube on the vortex mixer, the aqueous phase is added dropwise using a syringe with a needle approximately 12 cm in length, allowing it to reach the bottom of the tube. This configuration improves control over particle size and polydispersity index (PDI). The needle is kept perpendicular to the tube axis, and the drop rate is maintained as constant as possible until the entire aqueous phase has been added. During this step, the system transitions from an organic-phase-dominant mixture to a water-dominant mixture, resulting in phase inversion and the formation of the second oil-in-water emulsion.

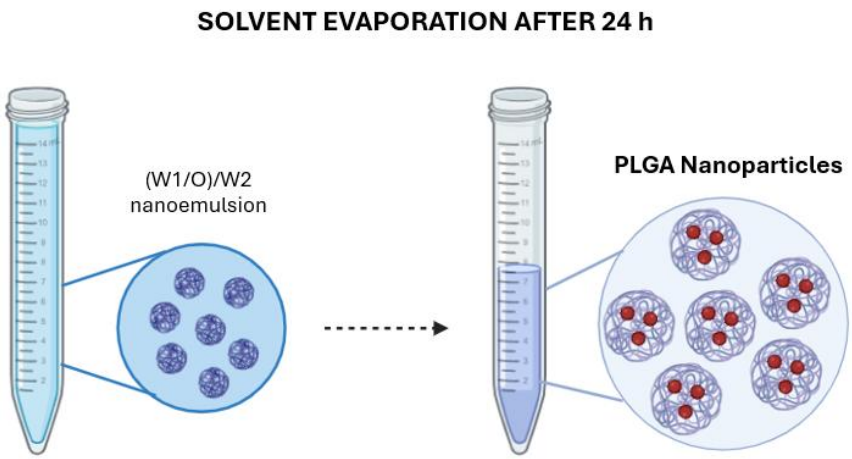


Figure 22 Solvent evaporation and nanoparticles formation

After forming the double emulsion, the Falcon tube is left uncovered for at least 24 h under a chemical fume hood at room temperature (22–25 °C) to allow complete evaporation of ethyl acetate.

3.2.2 Determination of phase inversion

In the drop-by-drop method, the transition from a water-in-oil (W/O) emulsion to an oil-in-water (O/W) emulsion can be observed. This phenomenon, known as phase inversion, is closely related to the relative proportions of the two phases. Initially, the oil phase is present in greater quantity, favoring the formation of a W/O emulsion. As water is progressively added, the proportion shifts, leading to the inversion of phases and the formation of an O/W emulsion.

To investigate this behavior, several mixtures of water and oil are prepared in 15 mL Falcon® tubes, with oil content ranging from 100% to 0% in increments of 10%. In each formulation, nanoparticles are produced using the drop-by-drop method, in which one phase is gradually added to the other under controlled agitation. The resulting emulsions are stored undisturbed for 60 days, allowing sufficient time for stabilization and for the phase inversion to become visually detectable.

To better identify the distribution of phases within the double emulsion, two parallel experiments are conducted. In the first experiment, the oil phase is labeled with Nile Red, a lipophilic dye dissolved in ethyl acetate and is added during the preparation of the first W/O emulsion. This labeling allows visualization of the oil phase under fluorescence microscopy.

In the second experiment, the aqueous phase is labeled with Rhodamine B, a water-soluble dye dissolved in Milli-Q® water and incorporated during the preparation of the second emulsion. In this case, fluorescence imaging enabled the tracking of the aqueous phase distribution.

The objective of this study is twofold: (i) to determine the precise point at which phase inversion occurs, and (ii) to visualize the continuous and dispersed phases within the emulsions using fluorescent markers.

3.2.3 Nanoparticles characterization

The characterization of a nanoparticle formulation requires the determination of several key parameters, including average size, polydispersity index (PDI), zeta potential (ζ-potential), encapsulation efficiency (EE%) and drug loading (DL%). The average size and PDI do not provide an exact particle diameter but rather describe the dimensional distribution of the particles in the sample. PDI values range from 0 to 1, and values above 0.4 indicate a polydisperse system in which particles vary widely in size. These parameters were measured using dynamic light scattering (DLS), a technique based on the analysis of light scattering caused by the Brownian motion of particles in suspension. In this method, a laser beam is directed at the nanoparticle suspension and the scattered light intensity is recorded over time. The fluctuations in intensity, which are more rapid for smaller particles and slower for larger ones, are analyzed through an autocorrelation function to obtain a size distribution and calculate the PDI.

Another important parameter is the zeta potential, which provides information on the surface charge of the particles and is a key indicator of colloidal stability. Although it is measured using the same instrument employed for DLS, the principle is different. An electric field is applied to the sample contained in a cuvette, causing charged particles to migrate towards the oppositely charged electrode. The electrophoretic mobility is measured and converted into a zeta potential value.

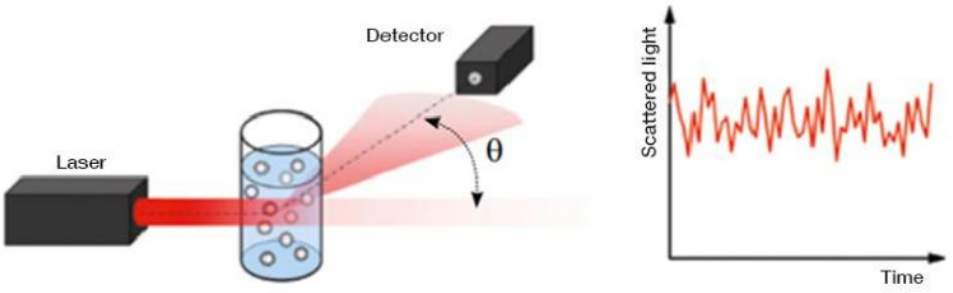


Figure 23 How Dynamic Light Scattering works for measuring nanoparticles characterization [74]

Encapsulation efficiency represents the proportion of drug incorporated into the nanoparticles compared to the initial amount used. In this work, Rhodamine B was employed as a model drug instead of doxorubicin due to its similar absorbance and logP values, at a concentration of 4.8 mg/mL. Following nanoparticle preparation, the suspension was separated from the non-encapsulated drug using Amicon® Ultra centrifugal filters with a 50 kDa cutoff. For each filtration, 500 µL of nanoparticle suspension were placed in the filter unit and centrifuged at 10,000 rpm for 25 minutes. The filtrate containing the unencapsulated drug was collected, and 100 µL of each sample were transferred in triplicate into the wells of a black 96-well microplate. Fluorescence measurements were performed using an M Plex microplate reader, and the concentration of Rhodamine B in the supernatant was determined using a previously established calibration curve. The encapsulation efficiency (EE%) was calculated from the difference between the initial drug mass and the free drug mass according to the formula:

$$EE\% = \frac{Mass\ initial\ drug - Mass\ free\ drug}{Mass\ initial\ drug} * 100 \quad (1)$$

Drug loading (DL%) indicates the mass of drug incorporated relative to the total mass of nanoparticles, thus providing a measure of the payload capacity of the system. It was calculated according to the formula:

$$DL\% = \frac{Mass\ encapsulated\ drug}{Mass\ nanoparticles} * 100 \quad (2)$$

This value is particularly important for assessing whether the formulation can deliver an adequate dose without excessive use of excipients.

3.2.4 Nanoparticles morphology

Cryo-TEM (cryo-transmission electron microscopy) is an extremely powerful technique for the direct observation of the morphology of nanoemulsions and nanoparticles. Unlike other microscopic techniques, this methodology allows you to visualize structures in conditions very close to their natural state, thanks to the rapid freezing of the sample which prevents the formation of ice crystals and preserves the integrity of the formulation. In the context of nanoemulsions, it allows to observe the droplet size, their distribution and the possible presence of aggregates; this information is essential for assessing the physical stability of the formulation and the behaviour of the system over time. It is possible to clearly distinguish dispersed droplets in the continuous phase and obtain more precise measurements than indirect techniques such as DLS. For nanoparticles, it allows the study of their shape, homogeneity and size distribution. It can also provide information on the presence of surface coatings or the formation of core-shell structures. This is particularly useful when working with complex systems, such as drug-loaded nanoparticles or functionalized surfaces.

Using a 2.5 mM PBS buffer solution, for both doxorubicin-loaded (DOX) and uncharged nanoparticles. A volume of 10 μL of the dispersion was deposited on a carbon coated copper grid (200 mesh) and left to stand for about one minute. The grates were then inserted into the FEI Vitrobot Mark IV vitrification system, where they were gently buffered to remove excess liquid and then quickly immersed in liquid nitrogen for instant freezing. The frozen grids were finally transferred to a cryogenic sample holder and observed by the Jeol 1400 transmission electron microscope. Image analysis of the captured pictures was performed using Image J software to determine the mean size of the nanoparticles.

3.2.5 Dataset construction

Machine learning models require an initial phase in which the variables, or features, characterizing the training examples are defined and selected. For the preparation of the datasets used in this work, several variables are chosen. The absolute volumes of water, oil, and surfactant are not used directly, as their conversion into percentage values makes the dataset both easier to interpret and more suitable for computational processing. Only results in which the sum of the percentages equals 100 are considered valid. This transformation also allows the compositions to be represented within a ternary diagram, a tool particularly effective for visualizing this type of data.

Each composition is then associated with a binary value (0 or 1) indicating the experimental observation of nanoemulsion formation (1) or its absence (0). The dataset also includes the values of particle size, polydispersity index (PDI), zeta potential (Z-potential), encapsulation efficiency (EE%), and drug loading (DL%), obtained either through direct experimental measurements or calculations. In cases where a composition does not produce an emulsion (class = 0), the values for these parameters are set to zero. For compositions located within a known region of the ternary diagram where nanoemulsions are formed but for which no measurements are available, the corresponding parameter values are assigned a value of −1, thereby distinguishing them as partially known cases while retaining the option to include them in classification training.

During the prediction phase, a filtering step is applied to exclude all entries with a water content below 50%. This choice is motivated by the objective of identifying oil-in-water (O/W) emulsions, which by definition require the aqueous fraction to be greater than the oil fraction. Additionally, this filtering helps to balance the dataset: an excess of negative cases (non-emulsions) can negatively affect model training, potentially causing it to overlook valid compositions or to suffer from overfitting.

Appendix A contains the complete table with the dataset used for the analysis.

3.2.6 Computational Methods for Data Analysis

The code is developed in Python (version 3.10), chosen as one of the most widespread and consolidated languages in the field of data science and industry. In addition to its wide adoption, Python offers a large set of machine learning-specific libraries and a simple, intuitive syntax, making it easy to use even in complex projects. Visual Studio Code is used as the development environment.

The NumPy library is used to manage the numerical data, fundamental for matrix and vector calculation on large datasets. The graphic representation of the experimental results and predictions is created with Matplotlib and Seaborn: the first allows highly customizable two-dimensional graphs to be built, while the second offers more immediate and aesthetically refined statistical visualizations. In addition, the python-ternary library, which is particularly suitable for this type of system, is used to represent the ternary compositions of the samples (water, oil, surfactant).

Analysis of machine learning models is conducted with the scikit-learn package. For classification, RandomForestClassifier is adopted, an algorithm based on sets of decision trees, useful for distinguishing compositions capable or not of generating nanoemulsions. For regression activities, GradientBoostingRegressor, which combines weak predictors through an incremental approach, and Support Vector Regressor (SVR), capable of capturing non-linear relationships between experimental variables, are used. In cases where it is necessary to predict multiple parameters simultaneously (for example mean size, polydispersion and zeta potential), MultiOutputRegressor is used, which extends regression algorithms to multivariate problems.

To ensure correct data management during the training phase, StandardScaler is adopted, which normalizes the variables on the same scale, improving the effectiveness of the models. The entire analysis workflow is organized via make_pipeline, which allows the preprocessing and modeling phases to be sequentially concatenated. Evaluation of the models is carried out using established procedures: train_test_split, to split the data into training and test sets, and cross_val_predict, which leverages cross-validation to get more reliable estimates and limit the risk of overfitting. Performance is quantified through numerical indices, specifically the Mean Squared Error (MSE) and the coefficient of determination R^2, indicators of the mean prediction error and the ability of the model to explain data variability, respectively.

Experimental data are loaded from a text file containing sample compositions and measured parameters. After an initial shuffling, the independent variables (water, oil and surfactant) are normalized as a percentage, imposing that their sum is equal to 1. The binary variable indicating the formation or not of a nanoemulsion is retained as the classification label, while the numerical output variables comprise average size, polydispersion index, zeta potential, encapsulation efficiency and drug loading magnitude.

A selective filter is also applied which excludes formulations with an aqueous fraction of less than 50%. This choice responds to the objective of the work, focused on oil-in-water (O/W) emulsions, which require a prevalence of the aqueous phase. In addition to this, the filter helps to rebalance the dataset, limiting the presence of overly numerous negative samples that could compromise the training of the models.

A Random Forest Classifier is used for classification, combining multiple decision trees trained in parallel. This approach is notable for its robustness to complex and noisy data, as well as reducing the risk of overfitting by mediating between multiple models.

For the regression of parameters such as size, polydispersion and zeta potential, a Support Vector Regressor (SVR) with basic radial kernel (RBF) is chosen, particularly suitable for capturing non-linear relationships and indicated for datasets of not very large dimensions. Finally, a Gradient Boosting Regressor is adopted for encapsulation efficiency (EE%) and drug loading (DL%), based on a sequence of incremental trees capable of modeling complex relationships and ensuring good predictive performance.

To comprehensively visualize the predictions, a ternary diagram is created that shows the regions of formation and non-formation of nanoemulsions: in green the compositions that lead to the formation of O/W emulsions, in red those that do not allow it. This type of representation offers both a global vision of the experimental space and an immediate comparison with any data already available.

Finally, an interactive prediction system is developed, which allows the user to manually enter a new trio of percentage values (water, oil and surfactant). The program checks the consistency of the data (sum of 100% and water ≥ 50%) and applies the classifier to establish whether the composition leads to the formation of a nanoemulsion. If positive, the model also provides a quantitative estimate of the main parameters: average particle size (Size), polydispersion index (PDI), zeta potential, encapsulation efficiency (EE%) and drug loading (DL%). In this way the system not only distinguishes between emulsions and non-emulsions, but also returns a complete characterization of the envisaged formulations

.

4. Results and discussion

In line with the objectives presented in the first chapter, the experimental work has as its main aim the definition of the optimal conditions for the formation of PLGA-based nanoemulsions and nanoparticles, through the construction of a ternary diagram representing the different combinations of water, oil and surfactant. This phase makes it possible to clearly delimit the regions of stability in which it is possible to obtain nanoemulsions, thus providing useful indications for the subsequent optimization of the formulations.

In parallel, the obtained nanoformulations are characterized in terms of size, polydispersity index (PDI), zeta potential, encapsulation efficiency and drug loading capacity. These parameters are critically analyzed to understand the impact of the different composition variables and identify the most significant trends for the purposes of designing safe and effective systems for drug delivery.

Finally, the computational part of the work introduces the application of machine learning models as a tool to support formulation. A first supervised model is developed to predict, starting from the composition, the probability of formation of a stable nanoemulsion, while a second framework allows estimating key properties of the nanoparticles (such as size, size distribution and encapsulation capacity) directly from the formulation variables.

In the following paragraphs, the main results obtained are presented and discussed, integrating experimental observations with computational predictions, with the final objective of identifying the most promising formulation to be employed in subsequent doxorubicin encapsulation studies for applications in oncology during pregnancy.

4.1 Nanoemulsion preparation

Before describing the preparation in detail, it is important to point out that the formation of a nanoemulsion is also assessed visually: after the preparation process, a transparent or slightly opalescent aspect of the dispersion is considered indicative of the presence of nanometric-sized droplets. In contrast, the appearance of marked turbidity or phase separation is interpreted as an index of non-formation of the nanoemulsion. Furthermore, if necessary, a few minutes are waited to allow the system to stabilize, in order to distinguish a real nanoemulsion from a simple temporary non-stable dispersion.

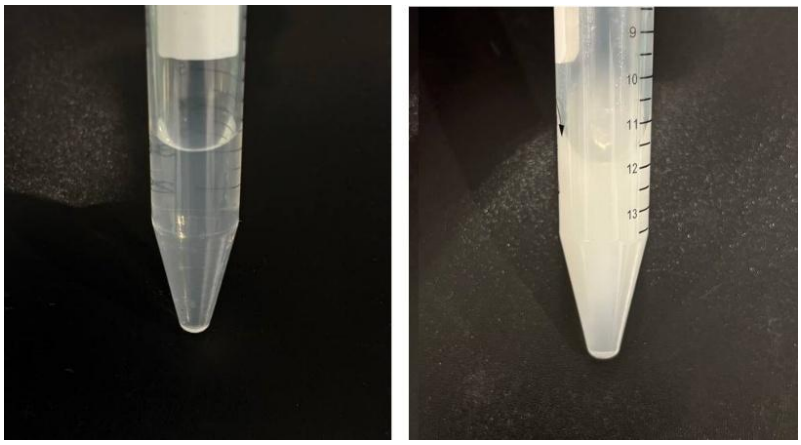


Figure 24 The picture on the left shows the formation of a nanoemulsion, as indicated by its transparent appearance; the picture on the right illustrates a system in which no nanoemulsion was formed.

Different combinations of water, oil and surfactant are tested during the preparation of the nanoemulsions. Only formulations in which the percentage of water is greater than 60% are taken into consideration, in order to obtain oil-in-water type systems. During the experiments, the ratios of surfactant to oil are varied, keeping the water range constant, so as to delimit the area of emulsion formation. The maximum limit of some regions is identified by comparing systems with the same percentage of water and proportional quantities of oil and surfactant, thus allowing the area of formation of nanoemulsions to be defined more precisely.

Starting from systems containing about 60% water, the addition of different amounts of oil does not lead to the formation of stable nanoemulsions. With the progressive increase in the aqueous phase, starting from values around 70%, the appearance of nanoemulsions begins to be observed, although not yet in optimal conditions from the point of view of surfactant content. This aspect is particularly relevant since, for reasons related to cell viability, a maximum surfactant content of less than 5% is placed as an experimental constraint.

By continuing to increase the aqueous fraction and modulating the quantities of oil and surfactant accordingly, it is possible to identify regions more favorable to the formation of nanoemulsions. In particular, starting from water values close to 88%, nanoemulsions are obtained even with minimal quantities of surfactant, less than 1%. This result is of particular interest, as it indicates the possibility of obtaining stable systems by drastically reducing the surfactant content, with clear advantages in terms of biocompatibility and potential applicability in the pharmaceutical/cosmetics sector.

Table 1 This table shows the formation of nanoemulsion (yes/no) at different percentage concentration of water/oil/surfactant (W/S/O)

Water %	Oil %	Surfactant %	emulsion yes/no	Water %	Oil %	Surfactant %	emulsion yes/no
60	39,5	0,5	no	87,5	11,5	1	yes
60	37,5	2,5	no	87,5	10	2,5	yes
60	35	5	no	87,5	7,5	5	yes
60	32,5	7,5	no	88	10,5	1,5	yes
60	30	10	no	89	10	1	yes
60	27,5	12,5	no	89	9	2	yes
70	29,5	0,5	no	90	9,5	0,5	yes
70	27,5	2,5	no	90	9,3	0,7	yes
70	25	5	no	90	9	1	yes
70	22,5	7,5	no	90	8,5	1,5	yes
70	20	10	no	90	8	2	yes
70	17,5	12,5	yes	90	5	5	yes
70	15	15	yes	91,2	8,3	0,5	yes
75	15	10	yes	91,2	7,8	1	yes
80	19,5	0,5	no	91,2	7,3	1,5	yes
80	17,5	2,5	no	92,5	7	0,5	yes
80	15	5	no	92,5	6,5	1	yes
80	12,5	7,5	no	93,5	5	1,5	yes
80	11,5	8,5	yes	94,5	4	1,5	yes
80	10	10	yes	95	4,7	0,3	yes
85	12,5	2,5	no	95	3,5	1,5	yes
85	11,5	3,5	yes	95	3	2	yes
85	10	5	yes	95	2,5	2,5	yes
85	7,5	7,5	yes	96	3,5	0,5	yes
86,5	11	2,5	yes	97,5	2,2	0,3	yes

From the results we obtain, we are able to identify and delimit a very precise region of the ternary diagram, within which stable emulsions are formed. This is an important aspect, because it allows us to limit the experimental conditions most favorable to the preparation of nanoemulsions, avoiding repetitive tests and clearly identifying the best ratios between water, oil and surfactant. The definition of this area also gives us the opportunity to compare our data with those reported in the literature, highlighting both similarities and differences, linked above all to the type of components used and the preparation method chosen. From an application point of view, being able to precisely identify the stability region is a fundamental step to develop reproducible and scalable formulations, as well as to optimize the quantity of surfactant used, which represents a crucial factor in terms of biocompatibility and safety.

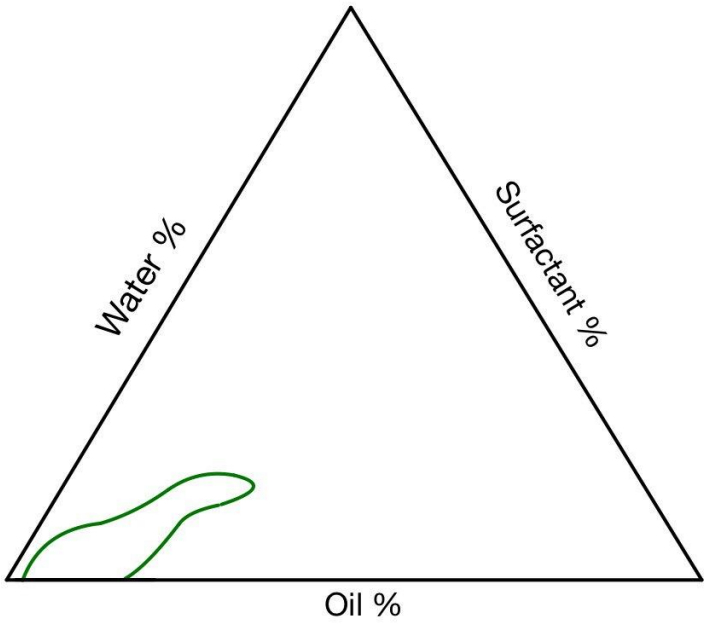


Figure 25 Ternary diagram where the green line identifies the emulsion region

4.2 Determination of phase inversion

The emulsions obtained show different characteristics depending on the oil/water ratio. For oil concentrations above 60% the system is milky and stable, with a typical W/O emulsion configuration. As the oil fraction decreases, and in particular with values lower than 50%, a progressive transition towards O/W type emulsions is observed.

After 60 days of storage it is possible to detect a marked phase separation in formulations containing less than 70% water, indicating a lower stability of the nanoemulsions in such conditions. In contrast, in preparations with an aqueous content of 70% or more (except for the 100% water sample, which lacks an oil phase), the system remains stable, with the surfactant still able to prevent visible separation of the phases even after the storage period.

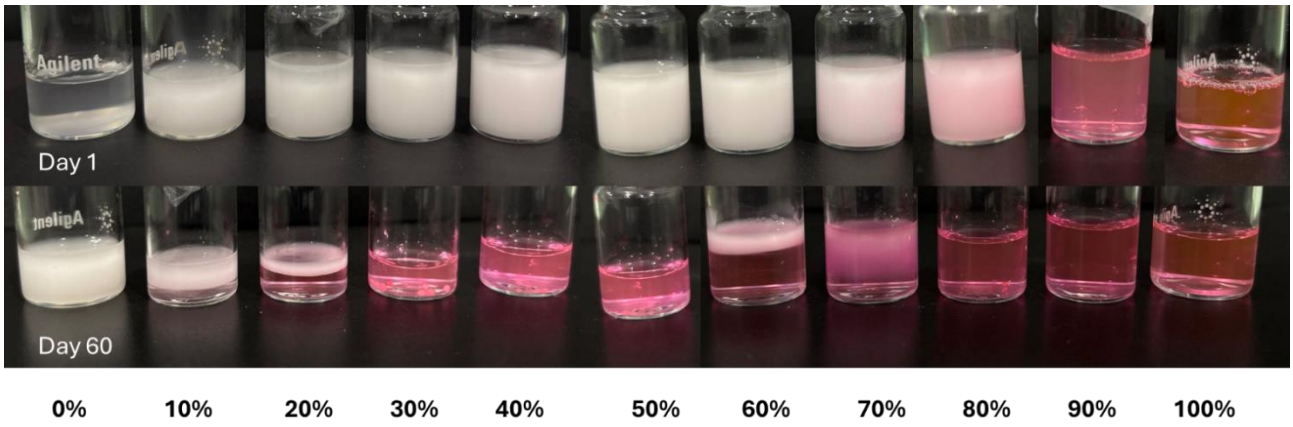


Figure 26 Phase inversion experiment with Rhodamine B: result after 60 days

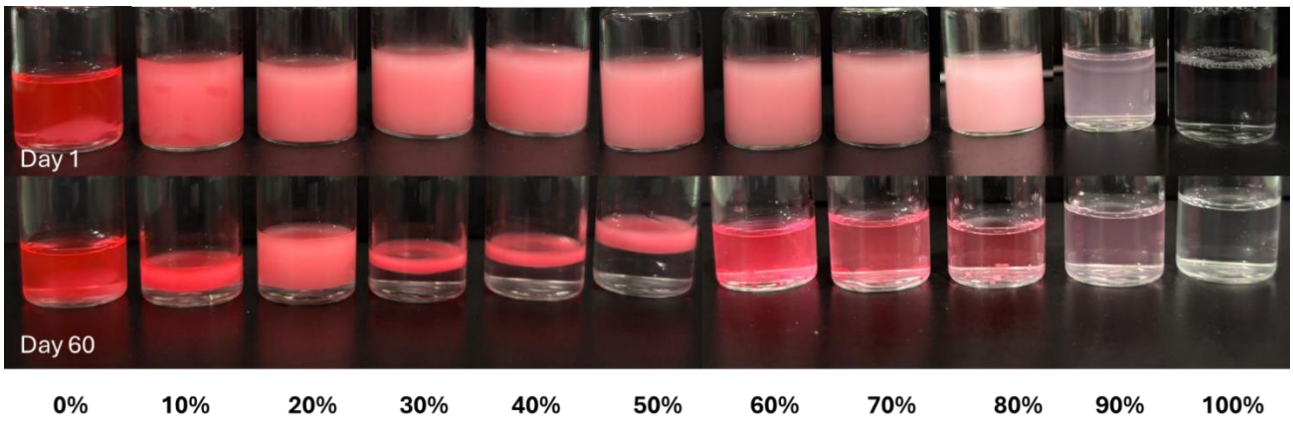


Figure 27 Phase inversion experiment with Nile Red: result after 60 days

The results obtained show how the relationship between the phases decisively influences the structure and stability of the emulsions. In formulations with high oil percentage (>60%), the formation of W/O systems is expected, since the continuous phase is determined by the majority component. Although these samples confirm the expected behavior, they are not the main objective of our study.

Instead, attention is directed to emulsions with higher aqueous content, in which progressive phase inversion towards O/W systems is observed. After 60 days of storage, preparations with less than 70% water show phase separation, pointing to limited stability over time. In contrast, formulations with water ≥70% (except the 100% oil phase-free control) maintain good stability, with the surfactant still able to prevent macroscopic separation phenomena.

These results suggest that there is a critical threshold around 70% aqueous phase beyond which the emulsion is longer lasting. This behavior is consistent with what is described for oil-water-

surfactant systems [30], where the prevalence of the continuous aqueous phase allows the surfactant to distribute the dispersed drops more efficiently, reducing coalescence.

4.3 Nanoparticles characterization

The characterization of nanoparticles represents a fundamental step, as it allows the key parameters of the system to be measured. These data are essential to evaluate their stability, quality and reproducibility.

Particle size provides insight into uptake and the ability to cross biological barriers. The polydispersity index (PDI) allows the uniformity and stability of the sample to be estimated. The zeta potential, on the other hand, is indicative of the colloidal stability of the nanoparticles.

Finally, the encapsulation efficiency and the drug loading make it possible to determine the amount of drug incorporated and to evaluate the effectiveness of the encapsulation process.

4.3.1 Dimension and polydispersity

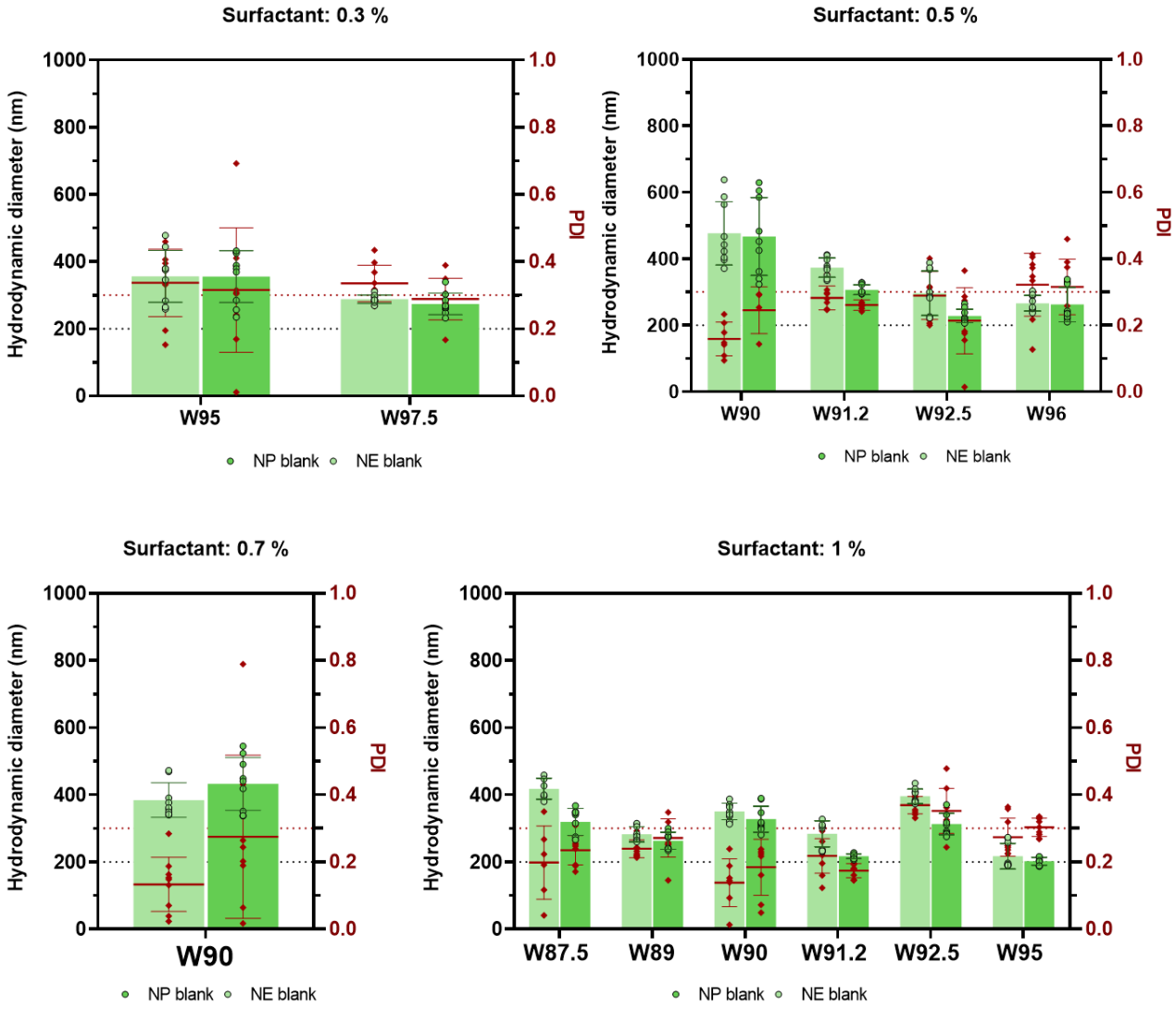
The mean diameter and polydispersity index (PDI) represent two closely related parameters, which can be assessed jointly to describe the quality of the nanoparticles. Analyzes conducted using Dynamic Light Scattering (DLS) have highlighted how the concentration of surfactant significantly influences both the size of the particles and their size distribution.

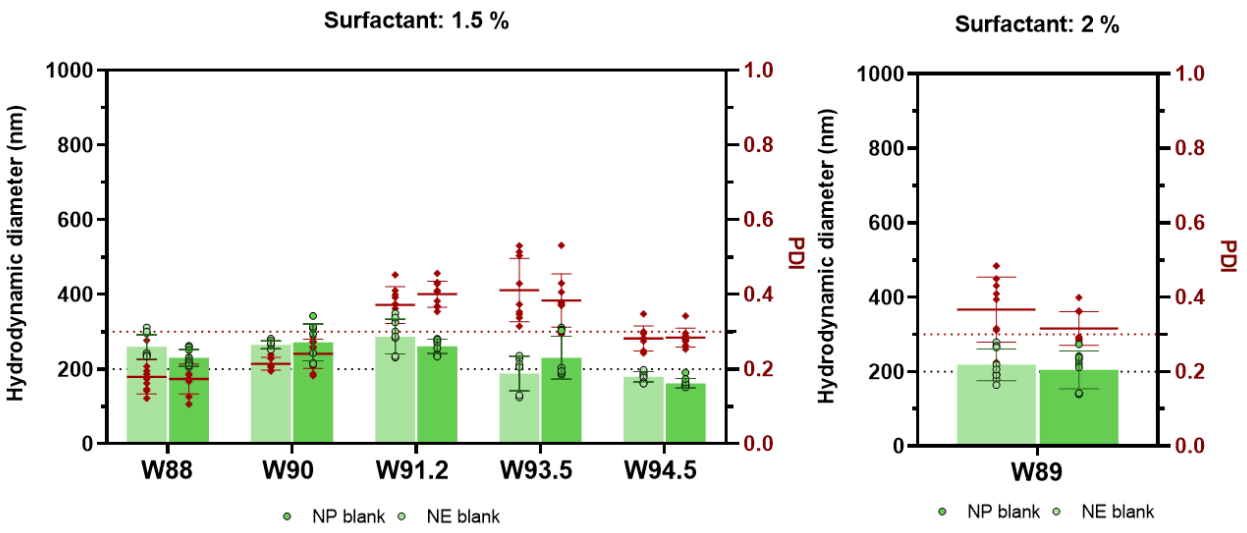
From the results, it is observed that, with surfactant concentrations lower than’1%, the nanoparticles have average diameters around 300 nm, a value that is at the upper limit of the range considered optimal for this type of system. Under these conditions, the PDI remains around 0.3, indicative of a relatively homogeneous distribution.

By increasing the concentration of surfactant up to approximately 3.5%, the particle size is progressively reduced, reaching values close to 200 nm, in this range 1% is considered optimal in the literature for bio compatibility in in vitro studies [75]. In parallel, the PDI shows good stability, oscillating between 0.2 and 0.4.

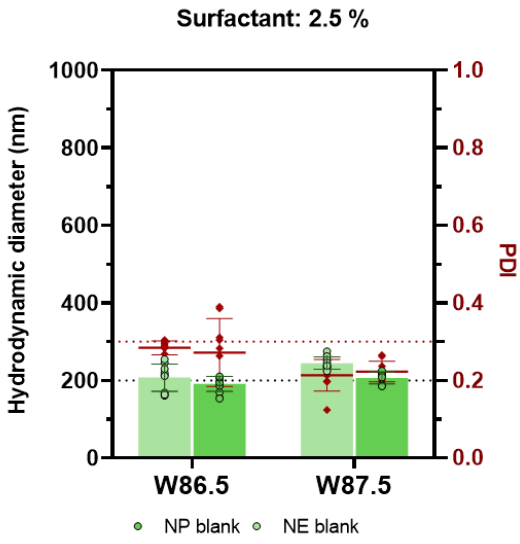
With surfactant concentrations equal to 5%, a critical behavior is highlighted: below a certain percentage of water the particles are stable, while over 90% of water there is a marked increase in PDI up to values around 0.6. This indicates a reduction in system stability, accompanied by an increase in size beyond 200 nm.

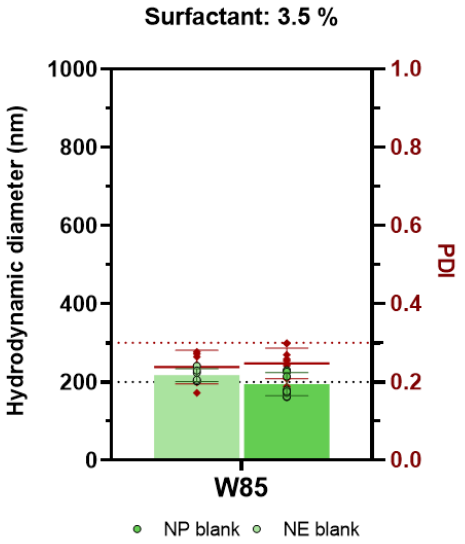
Once the 5% surfactant threshold is exceeded, the nanoparticles show a significant increase in size, which can even exceed 800 nm (as observed with 15% surfactant). Under these conditions, the PDI assumes high values, indicative of low colloidal stability and limited formulation reliability over time.

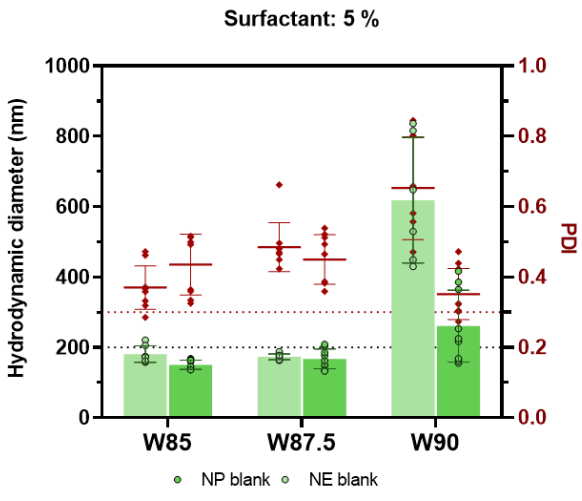


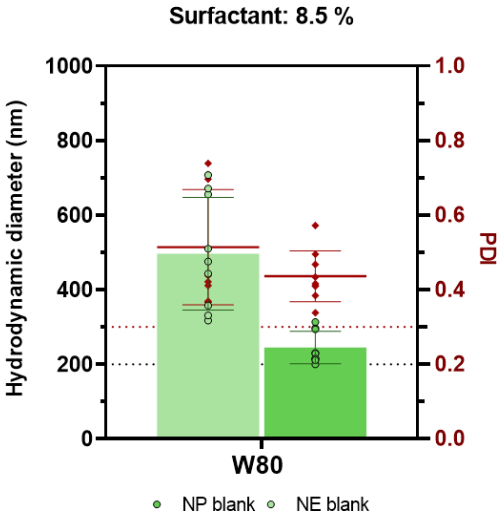

Surfactant: 1.5 %
Hydrodynamic diameter (nm)
PDI
W88
W90
W91.2
W93.5
W94.5
NP blank
NE blank

Surfactant: 2 %
Hydrodynamic diameter (nm)
PDI
W89
NP blank
NE blank


Surfactant: 2.5 %
Hydrodynamic diameter (nm)
PDI
W86.5
W87.5
NP blank
NE blank


Surfactant: 3.5 %
Hydrodynamic diameter (nm)
PDI
W85
NP blank
NE blank


Surfactant: 5 %
Hydrodynamic diameter (nm)
PDI
W85
W87.5
W90
NP blank
NE blank


Surfactant: 8.5 %
Hydrodynamic diameter (nm)
PDI
W80
NP blank
NE blank

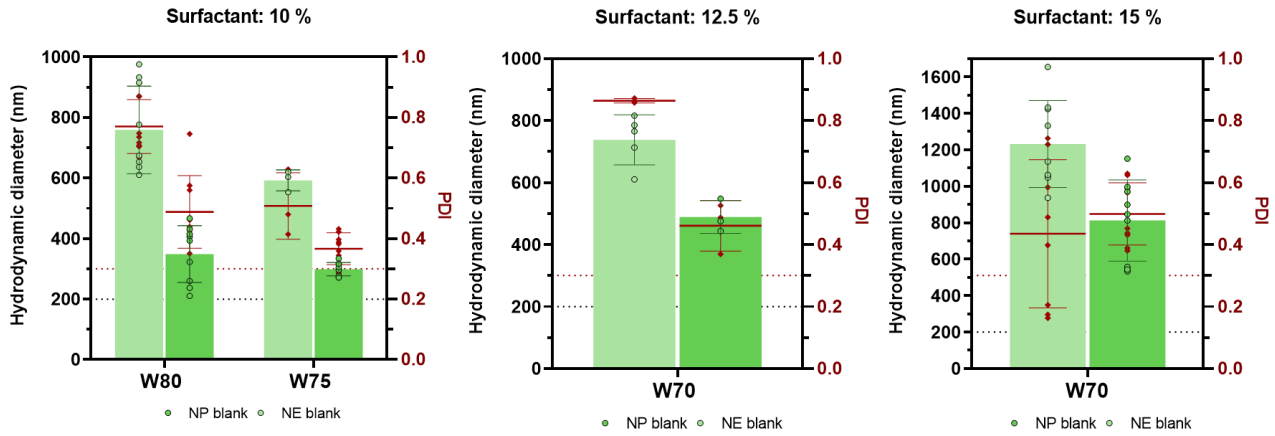


Figure 28 Series of graphs showing the evolution of the average size of nanoemulsions (light green), nanoparticles (green), and the polydispersity index PDI (red) as a function of surfactant percentage.

Data analysis highlighted how surfactant concentration significantly influences both the size and distribution of nanoparticles. In particular, concentrations lower than 1% do not allow sufficiently small diameters to be obtained, remaining around 300 nm. Intermediate concentrations, around 3.5%, allow values close to 200 nm to be reached, accompanied by good levels of uniformity (PDI 0.2–0.4), in accordance with what is reported in the literature. However, over 5% a progressive worsening of colloidal stability is observed, with an increase in PDI up to 0.6 and dimensions that can even exceed 800 nm, indicating the formation of unstable and poorly reproducible systems.

This analysis was also conducted with the aim of attributing a clearer meaning to the ternary diagram, outlining the regions where stable nanoemulsions can be obtained. The results obtained can be interpreted considering the interaction between surfactant molecules and the oil phase: at low concentrations the surfactant is not sufficient to stabilize the interface, while at too high concentrations the excess of molecules tends to generate instability phenomena (such as micellization or secondary aggregation).

From an application point of view, also considering the need to reduce the surfactant content to a minimum to preserve cell viability, the most promising data are those obtained with concentrations between 1% and 2.5%. This range represents an optimal trade-off between physical stability, molecular interactions and potential formulation biocompatibility.

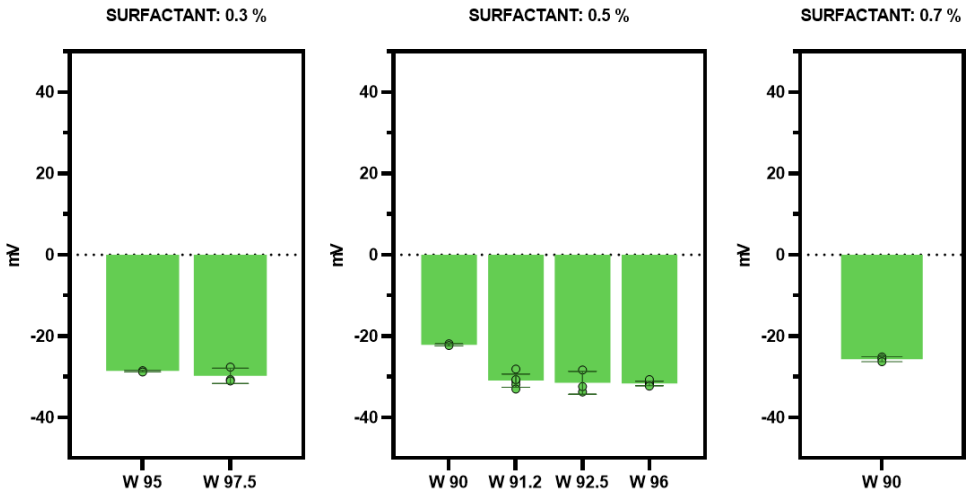
4.3.2 Z potential

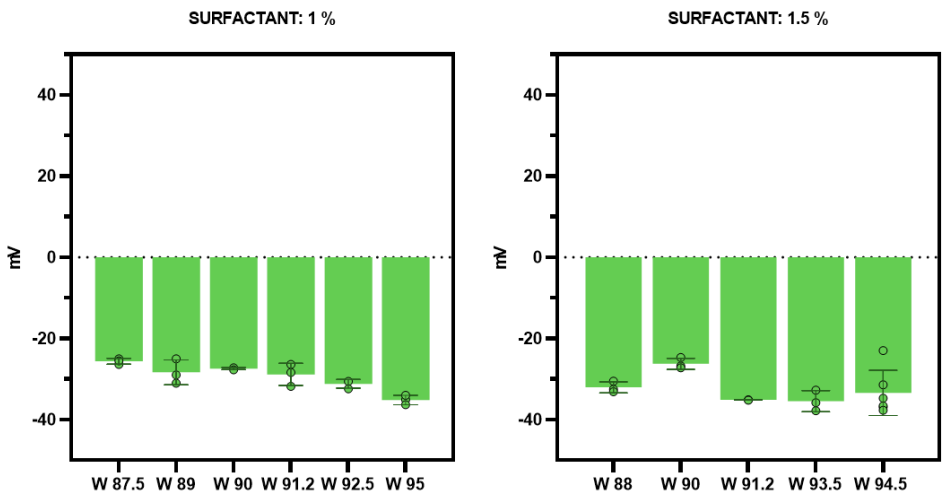
Another parameter that can be measured with Dynamic light scattering is the zeta potential, which represents the surface charge of the nanoparticles and is measured via electrophoresis.

This parameter is indicative of colloidal stability: high values (positive or negative) favor electrostatic repulsion and therefore stability, while values close to zero indicate tendency to aggregation. Furthermore, variations in the zeta potential can reveal any surface changes in the particles.

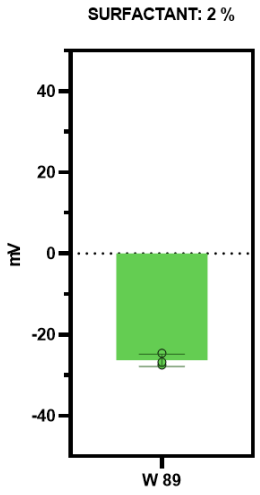
The measured zeta potential values are predominantly in the range of –20 to –35 mV. The results show that this parameter does not depend significantly on the concentration of surfactant, but on the ratio between water and oil: in particular, in all the graphs a tendency to increase (in absolute value) the zeta potential is observed with the increase in the percentage of water. The trend appears particularly clear in formulations containing ’1% surfactant, where the effects of different water concentrations are well defined.

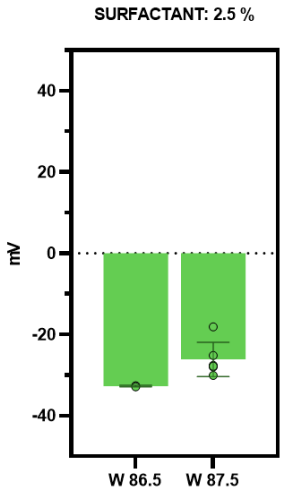
For surfactant concentrations above 5%, the zeta potential values stabilize around –30 mV, while the maximum recorded value, equal to approximately –40 mV, was observed only in the formulation with 15% surfactant.

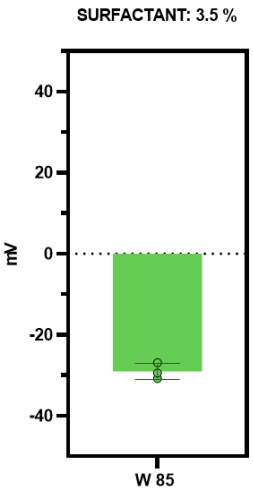


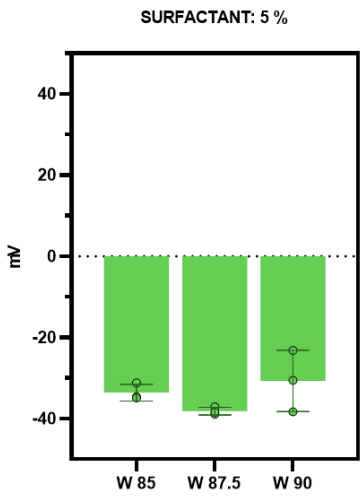

SURFACTANT: 1 %
mV
40
20
0
-20
-40
W 87.5
W 89
W 90
W 91.2
W 92.5
W 95

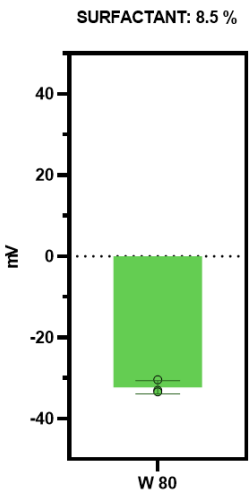
SURFACTANT: 1.5 %
mV
40
20
0
-20
-40
W 88
W 90
W 91.2
W 93.5
W 94.5


SURFACTANT: 2 %
mV
40
20
0
-20
-40
W 89


SURFACTANT: 2.5 %
mV
40
20
0
-20
-40
W 86.5
W 87.5


SURFACTANT: 3.5 %
mV
40
20
0
-20
-40
W 85


SURFACTANT: 5 %
mV
40
20
0
-20
-40
W 85
W 87.5
W 90


SURFACTANT: 8.5 %
mV
40
20
0
-20
-40
W 80

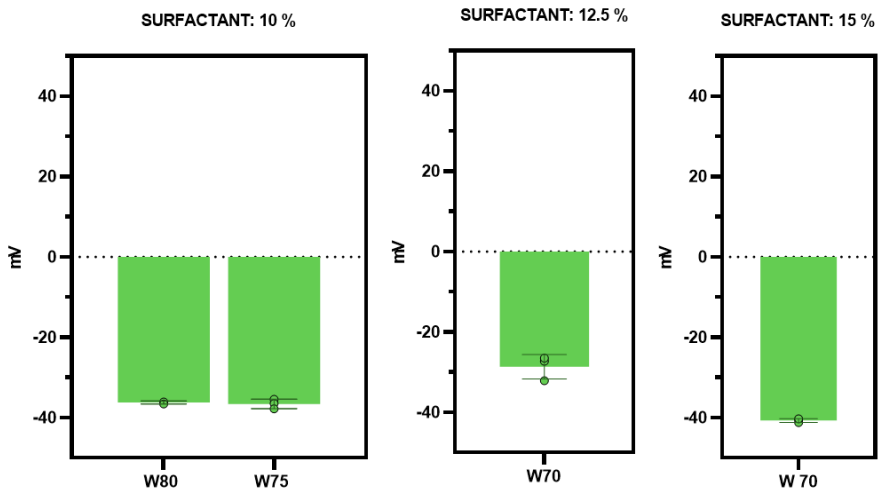


Figure 29 Series of graphs showing the trend of zeta potential as a function of surfactant percentage.

The data obtained show that the zeta potential values of the formulations remain stable overall, oscillating between –20 mV and –35 mV. This result is in line with what is reported in the literature (cite article), where values within the same range are described for PLGA-based nanoparticles. The absence of significant variations suggests that the system does not exhibit aggregation phenomena, thanks to sufficient electrostatic repulsion between the particles.

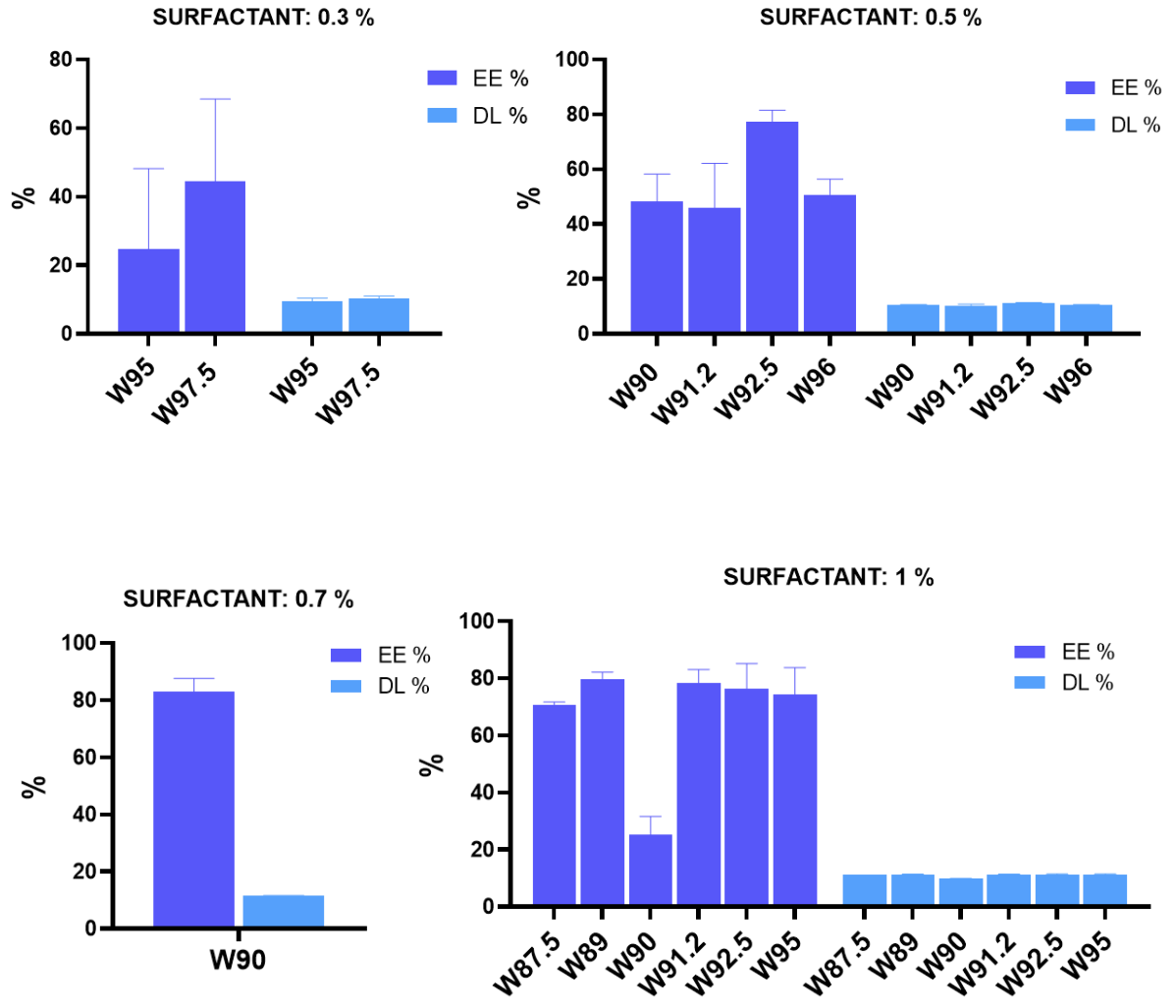
A relevant aspect that emerged from the analysis is that the zeta potential is not directly influenced by the concentration of surfactant, but mainly depends on the ratio between water and oil. When the surfactant concentration exceeds 5%, the values no longer change significantly but tend to stabilize around –30 mV. The only exception is the formulation with 15% surfactant, in which the zeta potential reaches about –40 mV. However, this result should not be considered advantageous: such a high content of surfactant compromises the biocompatibility of the system, limiting its applicability in the biological field.

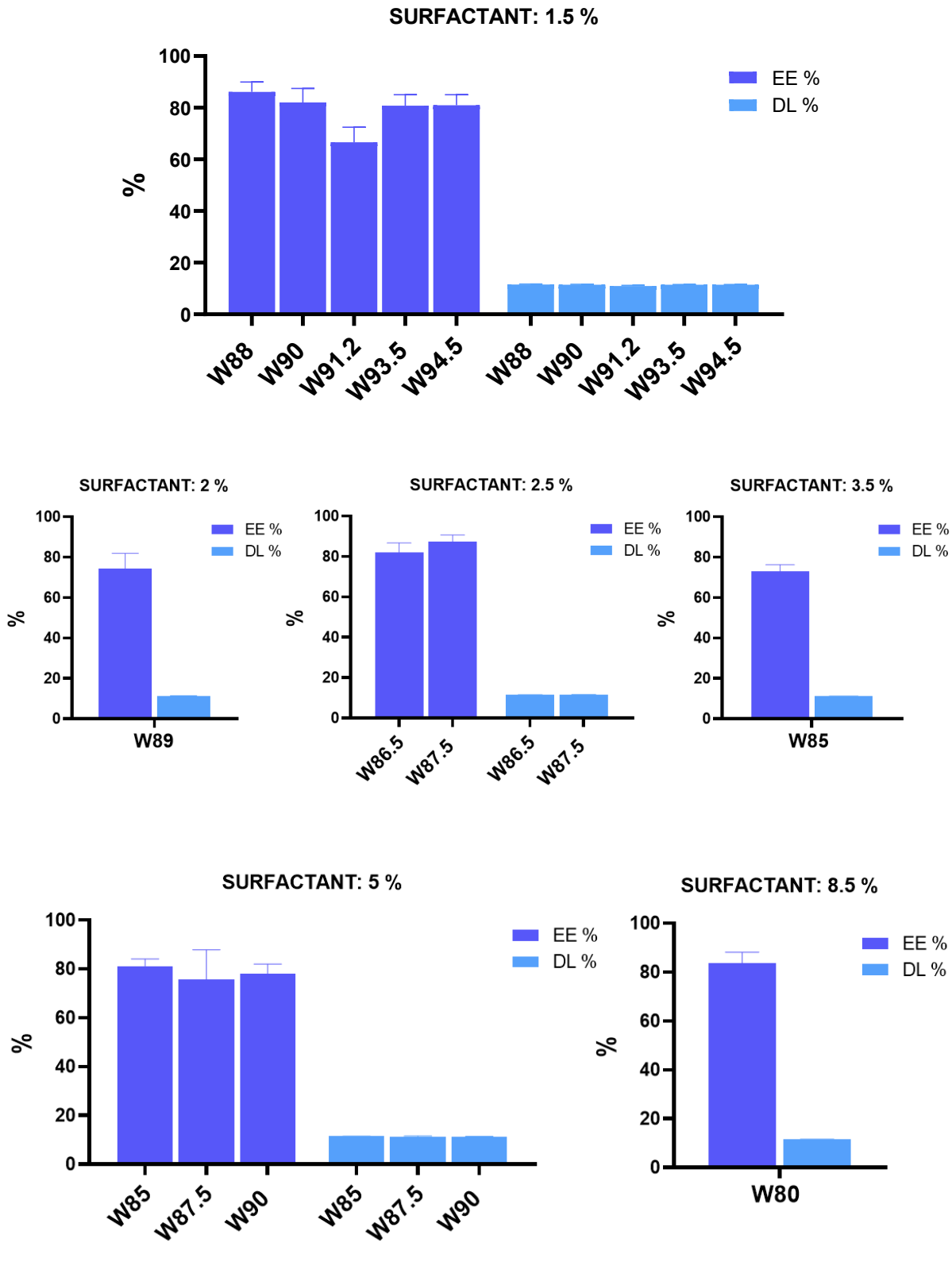
4.3.3 Encapsulation Efficiency and Drug Loading

Encapsulation efficiency (EE%) and drug loading (DL%) are two fundamental parameters in the study of nanoparticles and their interactions with drugs during the manufacturing process. In particular, the encapsulation efficiency expresses the amount of drug actually incorporated as a percentage of the amount initially used, thus providing a direct measure of the effectiveness of the system. Drug loading, on the other hand, indicates the amount of drug present inside the nanoparticles in relation to their overall weight, providing information on the ability of the vector to transport the active ingredient.

For surfactant concentrations below 0.7% it is observed how the encapsulation efficiency is often reduced: in several steps of variation of the aqueous fraction the yields are in fact kept below 60%, indicating that approximately half of the drug is not correctly incorporated into the nanoparticles. Regarding drug loading, the only value significantly below the norm was recorded in the formulation containing 0.5% surfactant and about 96% water, where the DL% falls below 10%.

By increasing the amount of surfactant, which is required for the formation of stable nanoemulsions, the encapsulation efficiency tends instead to stabilize around ’80%, which can be considered very satisfactory for a drug loading process. Drug loading also shows a more regular behavior: as the surfactant increases, the values remain around 15%, confirming the good ability of the system to incorporate the active ingredient.




SURFACTANT: 1.5 %
EE %
DL %
%
100
80
60
40
20
0
W88
W90
W91.2
W93.5
W94.5
SURFACTANT: 2 %
W89
SURFACTANT: 2.5 %
W86.5
W87.5
SURFACTANT: 3.5 %
W85
SURFACTANT: 5 %
SURFACTANT: 8.5 %
W80

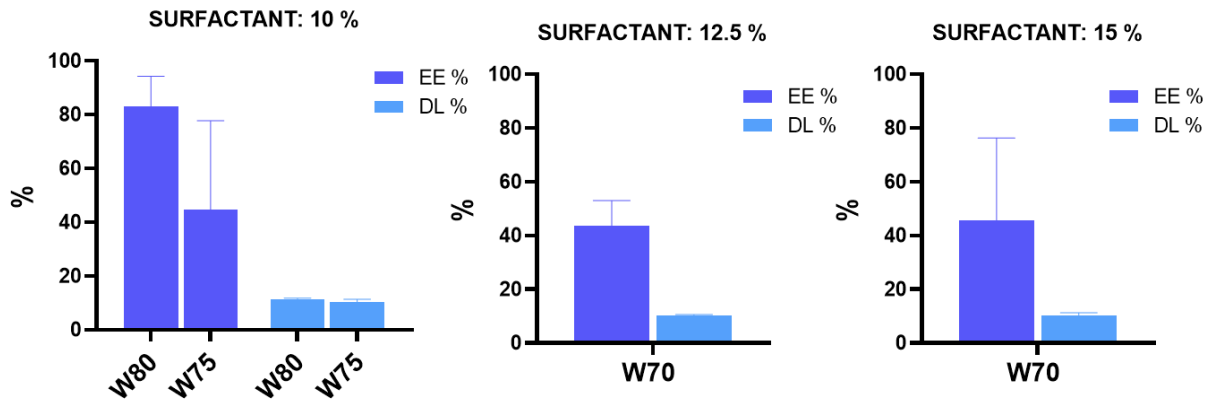


Figure 30 Series of graphs illustrating the evolution of encapsulation efficiency (blue) and drug loading (light blue) with increasing surfactant percentage.

Data analysis shows that, excluding formulations with low concentrations of surfactant (<0.7%), all systems examined present good encapsulation efficiency (EE%). In particular, EE% values above 70% indicate an effective loading process, with reduced waste of the active ingredient. However, it is essential to also balance drug loading (DL%): formulations with a DL% lower than 15% are uninteresting from an application point of view, as they would require high quantities of carrier to convey the therapeutic dose.

In the case under study, most formulations show DL% between 15% and 20%, confirming a good compromise between drug load and amount of excipients. In view of the objectives of this thesis, these results can be considered particularly promising, since they highlight the possibility of obtaining efficient and balanced systems, minimizing drug waste and reducing the use of surfactant in favor of better biocompatibility.

The obtained nanoparticles were characterized in terms of average size, polydispersity index (PDI), zeta potential, encapsulation efficiency (EE%) and drug loading (DL%). Overall, the data show good dimensions, with PDI values generally lower than 0.4, indicating good uniformity of the formulations. The zeta potential is maintained in the range of -25 and -35 mV, consistent with satisfactory colloidal stability.

Regarding the parameters related to the drug, the encapsulation efficiency is around 80%, while the drug loading values are mainly between 15% and 20%, confirming a good compromise between load capacity and quantity of excipients.

A cross-analysis of all the parameters allows us to identify as the most promising the formulations obtained with triples characterized by 91,2% of water and a concentration of surfactant that around 1%.

In fact, these combinations are distinguished by the formation of stable nanoemulsions, small and uniform dimensions, adequate zeta potential values and, above all, an optimal balance between encapsulation efficiency and drug loading, without exceeding the use of the surfactant, a fundamental aspect for biocompatibility.

4.4 Nanoparticle morphology

In the evaluation of the morphological characteristics, we can certainly confirm the formation of spherical nanoparticles and that both in the case of empty nanoparticles and in the case of nanoparticles loaded with doxorubicin we have a size contained below 250 nm.

This result leads us to say that even when loading the drug, the dimensions of the nanoparticles do not alter, leaving the range of the empty nanoparticle.

Furthermore, maintaining the same spherical structure in the charged nanoparticles indicates that drug incorporation does not compromise formulation stability.

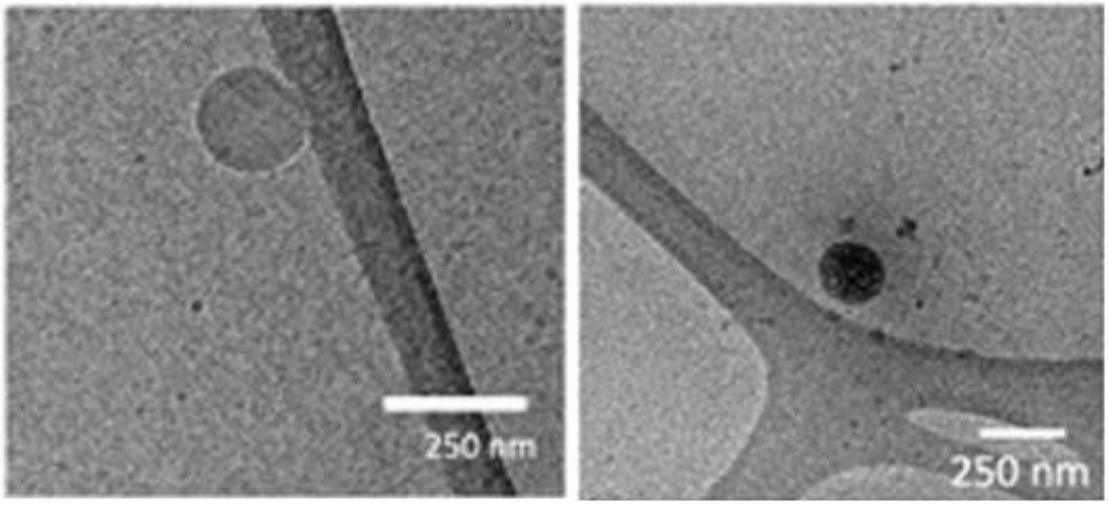


Figure 31 Visualization of nanoparticle with cryo-TEM: in the left side without drug loaded; in right side loaded with DOX

4.5 Ternary map for nanoemulsion prediction

The prediction of emulsions is carried out via a Random Forest classifier. The strength of this method lies in the combination of the results of multiple decision trees, each contributing to a more robust and less noise-sensitive final decision in the data, a feature that is particularly useful in the analysis of complex models.

Among the different configurations tested, a model with 42 trees is chosen, as it provides the best performance, with an accuracy of approximately 82%. To ensure a reliable evaluation, the dataset is divided into a training set (80%) and a test set (20%), thus reducing the risk of overfitting and allowing the predictive capacity of the model to be estimated more realistically.

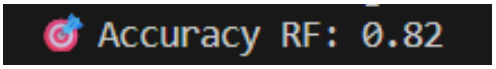


Figure 32 Result of accuracy calculated by the python in Visual Studio while running the code for nanoemulsion prediction

The measure used to evaluate performance is accuracy, which represents the proportion of correctly classified observations.

Subsequently, the predictions are represented on the ternary diagram, so as to analyze the distribution of nanoemulsions. The points relating to the nanoemulsion formulations predicted by the model (or already present in the dataset) are shown in green, while the areas in which the formation of nanoemulsions is not observed are indicated in red. This representation allows a clear visualization of the areas of stability and instability of the studied system.

The code integrates an interactive section that, when entering a new triplet, assesses the feasibility of nanoemulsion formation at that specific point.

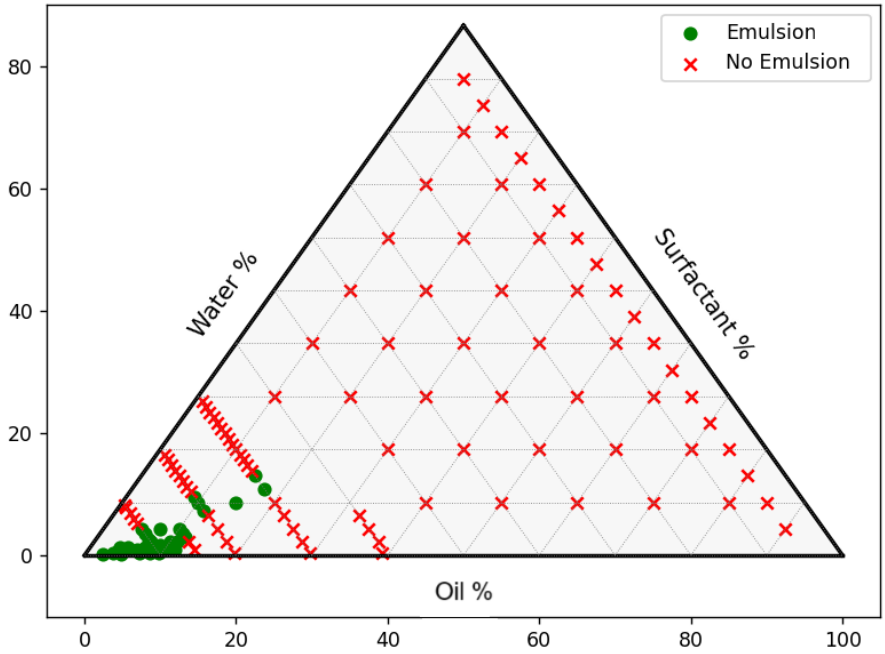


Figure 33 First ternary diagram that the compiler give as result of nanoemulsion prediction: we have a positive result of nanoemulsion (green dots) and negative results of nanoemulsion formulation (red cross)

Subsequently, the values reported on the ternary diagram are thickened taking into account the different regions, in order to build a distribution map. This map allows us to clearly distinguish the areas in which the formation of nanoemulsions is expected from those in which, on the contrary, their formation does not occur. In this way it is possible to define the zones of stability and instability of the system more precisely.

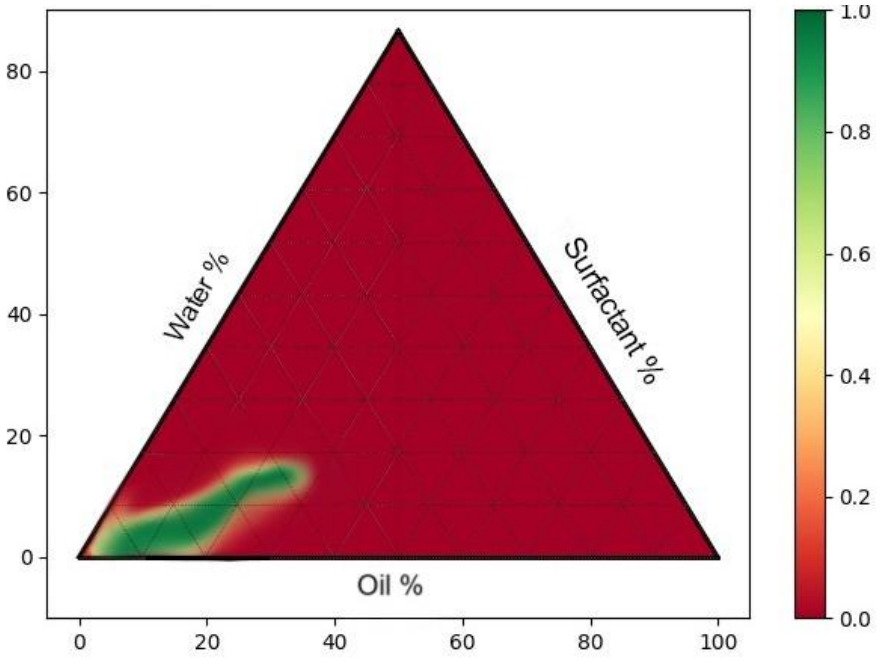


Figure 34 Distribution of prediction nanoemulsion in the ternary diagram; in green the formulation of nanoemulsion; in red there isn't nanoemulsion formulation

From the results obtained it can be stated that the choice of the number of trees and the subdivision of the dataset represent an effective strategy to evaluate the predictive capabilities of the model, also taking into account the combinations considered in the construction of the dataset and reducing the risk of overfitting.

The Random Forest model is therefore confirmed as a reliable solution for data prediction, as also demonstrated by the distribution map, which allows a clear visualization of the areas of presence and absence of nanoemulsions. This result emphasizes the robustness of the model and its practical utility in the analysis and prediction of the behavior of the studied system.

4.6 Analysis of Computational Result

Computational analyses are conducted by inserting the dataset into the different codes, each based on different machine learning methods (K-Nearest Neighbors, Support Vector Regression, Gradient Boosting, Polynomial Regression, Random Forest), with the aim of predicting a specific composition triplet. The predictions are then verified experimentally in the laboratory, in parallel, by comparing the values obtained with those predicted and calculating the percentage deviation between the two.

The K-Nearest Neighbors (KNN) algorithm is appreciated for its simplicity and ability to adapt to nonlinear data without particular assumptions. Support Vector Regression (SVR), thanks to

the kernel trick, is particularly effective in capturing complex relationships and is robust in the presence of outliers. Gradient Boosting-based methods are distinguished by high accuracy, achieved by combining multiple weak models into a powerful predictor. Polynomial regression, on the other hand, represents a direct and intuitive approach to modeling nonlinear relationships while maintaining the simplicity of linear regression. Finally, Random Forests guarantee robustness with respect to noise and the risk of overfitting, thanks to the construction of a multitude of decision trees and their aggregation.

This prediction approach is developed by initially considering each variable independent of the others (single-variant approach) and subsequently the possible interactions between the various parameters (multivariant approach). To this end, the data are organized into two distinct groups, according to the type of measurement used: on the one hand, DLS, which provides multivariate parameters such as size, PDI and zeta potential; on the other hand, the plate reader, which takes into consideration encapsulation efficiency and drug loading.

Tables 2 and 3 show the relative prediction errors, referring to the single-variant approach, and allow us to understand how the performance of the models differs appreciably. In particular, Gradient Boosting and Random Forest are the techniques with the lowest relative errors, confirming their effectiveness in capturing complex patterns and reducing the influence of noise or outliers. Support Vector Regression (SVR) is also positioned among the most accurate models, thanks to the ability of the kernel to adapt to nonlinear relationships. In contrast, K-Nearest Neighbors (KNN) and Polynomial Regression show higher mean errors; in particular, the latter highlights difficulties in handling extreme values and more complex variations.

In the single-variant context, the analysis confirms that models based on more complex approaches offer more precise predictions, while simpler approaches are less robust when faced with nonlinear patterns or data variability.

Table 2 Table of relative error for size, PDI and Z potential in single variant approach

	RELATIVE ERROR														
tern W/O/S %	SIZE					PDI					Z POTENTIAL				
	GB	KNN	PR	SVR	RF	GB	KNN	PR	SVR	RF	GB	KNN	PR	SVR	RF
91,2/7,3/1,5	37,18	32,36	26,29	1,26	10,44	40,76	53,15	44,53	25,69	58,54	10,54	46,72	41,03	13,96	16,81
89/9/2	22,49	35,39	12,90	61,95	27,21	49,54	61,27	50,35	18,99	59,08	3,80	38,02	16,35	10,27	3,80
93,5/5/1,5	170,86	51,13	0,38	55,12	133,93	8,10	42,13	35,81	25,12	20,50	9,12	12,50	35,37	15,61	10,81
86,5/11/2,5	4,39	0,39	0,68	53,91	11,33	23,67	6,44	52,87	11,15	32,82	21,10	29,36	29,66	17,43	17,43
88/10,5/1,5	32,03	44,31	99,87	4,49	45,55	12,85	1,68	0,56	61,45	13,41	17,58	16,95	19,76	20,07	16,64
94,5/4/1,5	103,69	82,06	100,96	80,50	39,98	29,79	21,63	50,71	9,22	33,33	1,98	7,48	25,51	8,37	0,09
91,2/8,3/0,5	9,21	15,11	12,24	5,40	15,77	43,95	8,48	32,60	5,71	29,76	14,32	7,21	16,26	12,71	12,06
96/3,5/0,5	33,76	58,35	23,34	39,14	28,61	4,76	17,81	10,66	12,03	1,15	6,70	0,57	12,08	3,10	5,12
89/10/1	33,76	58,35	23,34	39,14	28,61	4,76	17,81	10,66	12,03	1,15	14,93	5,75	3,64	9,99	7,52
95/4/1	63,25	48,72	121,57	70,58	61,17	16,10	12,81	74,52	6,61	18,29	14,32	9,48	22,29	5,78	12,61

Table 3 Table of relative error for encapsulation efficiency and drug loading in single variant approach

tern W/O/S %	RELATIVE ERROR									
	EE %					DL %				
	GB	KNN	PR	SVR	RF	GB	KNN	PR	SVR	RF
91,2/7,3/1,5	18,51	30,99	30,24	20,91	17,01	2,70	42,34	34,23	3,60	1,80
89/9/2	9,83	34,99	24,09	7,81	8,61	1,51	42,12	26,98	3,29	1,51
93,5/5/1,5	11,30	22,68	50,64	15,01	13,15	1,93	4,55	34,33	12,43	2,80
86,5/11/2,5	1,11	23,70	17,35	8,92	2,45	0,52	22,34	14,49	0,52	0,52
88/10,5/1,5	10,57	15,80	14,52	11,03	7,32	2,42	3,28	2,42	2,42	1,55
94,5/4/1,5	53,56	27,75	47,02	19,85	35,04	11,64	5,51	26,51	1,14	8,14
91,2/8,3/0,5	40,70	8,22	24,70	7,99	30,46	9,09	2,16	13,42	2,16	6,49
96/3,5/0,5	6,01	16,65	7,78	9,75	4,63	141,22	152,93	120,14	150,59	143,56
89/10/1	3,18	6,84	4,58	1,58	0,08	0,88	1,75	0,88	0,00	0,00
95/4/1	56,45	26,92	34,61	17,35	19,78	60,82	16,30	26,98	5,61	15,41

Tables 4 and 5 show the relative prediction errors for the various models in the multivariant approach, considering different output variables (SIZE, PDI, Z POTENTIAL, EE %, DL %). Overall, it emerges that Gradient Boosting (GB) and Random Forest (RF) tend to keep relative errors lower in most cases, confirming their ability to effectively manage the complexity introduced by multiple simultaneous predictors. Support Vector Regression (SVR) shows competitive performance, especially for some variables (e.g. SIZE and PDI), demonstrating its effectiveness in modeling nonlinear relationships even in multivariant contexts.

In contrast, K-Nearest Neighbors (KNN) and polynomial regression (PR) tend to show higher errors in many cases, especially when the number of variables increases or when the data has extreme values or strong variability. Interestingly, some variables, such as DL %, are predicted with overall very low errors by all models, while others, such as EE %, highlight greater variability in errors, suggesting greater complexity in their prediction.

In summary, even in the multivariant context, ensemble-based algorithms confirm their superiority in terms of predictive accuracy, while simpler methods are less robust in the face of additional complexity due to the presence of multiple simultaneous predictors.

Table 4 Table of relative error for size, PDI and Z potential in multivariant approach

tern W/O/S %	RELATIVE ERROR														
	SIZE					PDI					Z POTENTIAL				
	GB	KNN	PR	SVR	RF	GB	KNN	PR	SVR	RF	GB	KNN	PR	SVR	RF
91,2/7,3/1,5	2,63	7,93	31,45	1,49	12,89	41,84	42,11	53,96	43,46	49,65	10,54	17,38	14,25	13,96	21,65
89/9/2	22,49	8,32	5,63	22,05	38,72	59,36	69,72	47,90	70,54	66,18	3,80	3,04	12,17	10,27	3,04
93,5/5/1,5	39,60	40,82	5,43	57,83	138,92	8,34	41,16	31,92	36,79	6,98	9,12	12,50	11,94	15,61	11,37
86,5/11/2,5	23,43	5,74	31,15	27,68	0,29	26,13	17,34	22,27	15,58	30,71	21,10	14,07	18,96	17,43	16,82
88/10,5/1,5	37,30	37,77	24,59	10,07	47,20	7,82	2,23	7,82	12,29	7,82	17,58	16,95	17,89	20,07	14,77
94,5/4/1,5	113,39	79,83	98,12	93,21	39,19	29,43	9,57	48,23	59,93	35,11	1,98	7,48	7,48	8,37	0,38
91,2/8,3/0,5	19,93	5,03	14,19	12,48	16,23	41,82	29,05	44,31	6,70	20,18	14,32	7,21	7,53	12,71	5,59
96/3,5/0,5	10,75	62,83	5,90	46,17	25,15	4,13	20,92	10,04	48,27	1,65	6,70	0,57	1,96	3,10	5,76
89/10/1	10,75	62,83	5,90	46,17	25,15	4,13	20,92	10,04	48,27	1,65	14,93	5,75	4,34	9,99	10,34
95/4/1	56,84	48,72	43,61	67,58	73,62	14,28	12,81	40,93	73,42	26,32	12,33	9,48	10,33	5,78	12,33

Table 5 Table of relative error for encapsulation efficiency and drug loading in multivariant approach

	RELATIVE ERROR									
tern W/O/S %	EE %					DL %				
	GB	KNN	PR	SVR	RF	GB	KNN	PR	SVR	RF
91,2/7,3/1,5	18,51	18,21	5,01	20,91	17,46	2,70	2,70	0,00	3,60	1,80
89/9/2	9,83	2,56	1,35	7,81	8,88	1,51	0,62	0,62	3,29	1,51
93,5/5/1,5	11,30	22,68	25,89	15,01	14,14	1,93	4,55	5,43	12,43	2,80
86,5/11/2,5	1,11	5,26	5,75	8,92	5,75	0,52	1,40	1,40	0,52	1,40
88/10,5/1,5	10,57	15,80	12,20	10,92	6,50	2,42	3,28	2,42	2,42	1,55
94,5/4/1,5	52,20	27,75	29,23	19,85	35,16	11,64	5,51	6,39	1,14	7,26
91,2/8,3/0,5	40,70	8,22	16,58	7,99	28,82	9,09	2,16	3,90	2,16	6,49
96/3,5/0,5	15,07	16,65	6,80	9,75	10,54	141,22	152,93	148,24	141,22	143,56
89/10/1	3,18	6,84	4,71	1,58	1,08	0,88	1,75	0,88	0,00	0,00
95/4/1	56,45	26,92	20,86	17,35	16,27	60,82	16,30	13,62	5,61	14,51

In order to comprehensively evaluate all the tables containing the relative errors, a multivariate approach for predictions was implemented in the code, which proved to be particularly effective in capturing the internal relationships between the variables.

Furthermore, for the two sections of code dedicated to predictions, a Support Vector Regression (SVR) approach was adopted to estimate the data relating to the size, the PDI measurement and the z potential, by virtue of the good performance reported. For the prediction of variables related to encapsulation efficacy and drug loading, the Gradient Boosting method was employed, chosen for its consistency and for the low maintenance of relative errors.

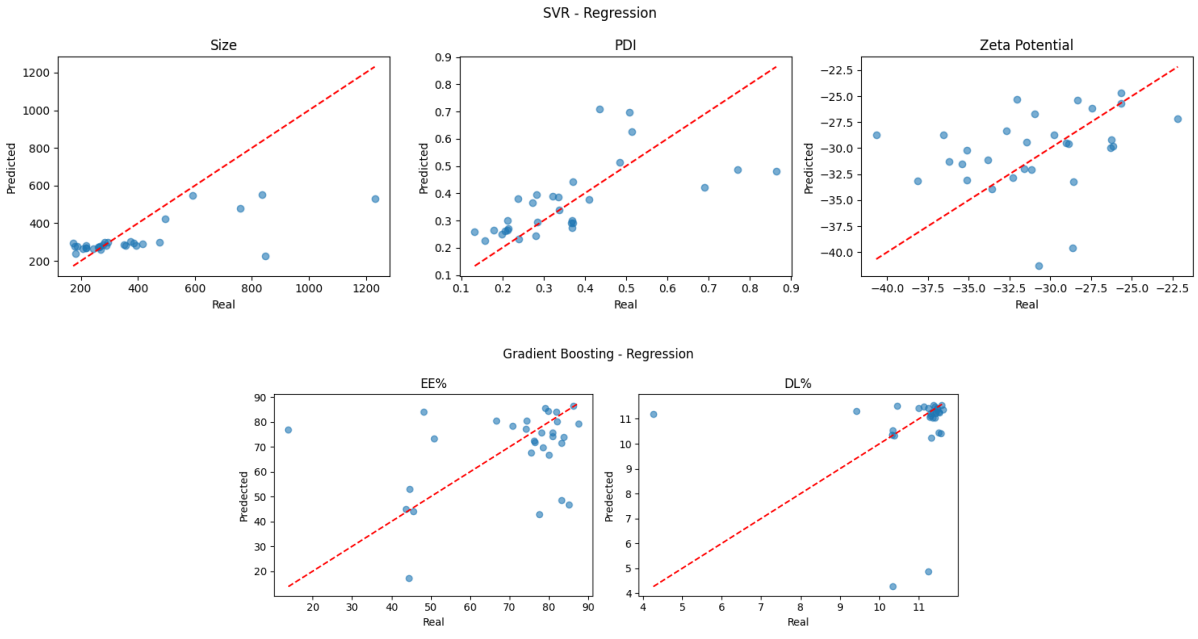


Figure 35 Visual representation of the accuracy of both regression model. Each blue point has coordinates corresponding to the real value (x-axis) and the predicted value (y-axis). The red dashed line represents the bisector (y = x): the closer the points are to this line, the higher the model's accuracy.

As we can observe from the graphs, the models show the points corresponding to the real and predicted values in blue. The closer these points are to the bisector, the greater the precision of the model. The only case in which greater dispersion is highlighted concerns the graph relating to the zeta potential. However, this aspect does not represent a significant criticality: in fact, when considering the potential of PLGA alone, this value tends to oscillate within a well-defined range. In the absence of additional external agents that can modify its potential, the observed variability does not compromise the overall reliability of the predictions.

Appendix B provides the final Python code developed for this work.

5. Conclusions

The results obtained in this thesis lead to several significant conclusions.

The first objective of the work concerned the formulation of nanoemulsions and subsequent nanoparticles, with the aim of identifying the ideal characteristics for future studies. The experimental results showed that the nanoparticles obtained with a composition of 91.2% water and all'1% surfactant showed the best performance in terms of critical parameters such as size, polydispersity index, zeta potential, encapsulation efficiency and drug loading.

Such properties make these nanoparticles a promising vehicle for targeted treatment of pregnancy-associated breast cancer, with the potential benefit of avoiding passage through the placental barrier.

At the same time, computational objectives led to equally satisfactory results. In fact, a machine learning algorithm has been developed capable, starting from a dataset of previously calculated nanoemulsion triples, of generating a ternary diagram including all the obtainable nanoparticles.

Moreover, the system allows to interactively recognize if new combinations inserted by the user lead to the formation of further nanoemulsions. For this purpose, a supervised algorithm of the Random Forest type was used, which showed good efficacy with an accuracy of more than 75%.

A further objective was to extend the predictive capability of the algorithm, including the main characteristics of the nanoparticles (size, polydispersity index, zeta potential, encapsulation efficiency and drug loading). The results highlighted that, for size, polydispersity and zeta potential, the most accurate algorithm was the Support Vector Regressor, while for encapsulation efficiency and drug loading the model with the best performance was the Gradient Boosting. This demonstrates how the computational approach allows optimizing the design of nanoparticles, reducing time, costs and material waste in experiments.

Finally, the conclusive objective concerned the identification of the most promising nanoparticle for future studies with doxorubicin encapsulation. The combination with 91.2% water and '1% surfactant was confirmed as the one with the most favorable characteristics, proving not only effective, but also biocompatible.

The formulation identified as optimal has a particularly advantageous balance between stability and load capacity, which represents added value. Furthermore, the prospect of applying such

nanoparticles to pregnancy-associated breast cancer is of great clinical relevance, as currently available treatment options are limited and often problematic for foetal safety.

The results obtained during this work made it possible to respond to the set objectives, both in the experimental and computational parts. However, it is important to focus critically on the meaning of these data, the limits they present and possible future prospects.

As regards the experimental part, the lack of validations through in vitro and subsequently in vivo tests represents a limitation that will have to be addressed. Future studies, conducted on more complex biological models, could in fact confirm or improve the observed performance, allowing the characteristics of the nanoparticles to be further optimised.

Computationally, the employment of machine learning algorithms showed good predictive ability, with overall satisfactory levels of accuracy. However, the performance of the models could be further improved by dataset expansion and enrichment, including more numerous and diverse combinations.

An innovative aspect of the work lies precisely in the integration between experimentation and computational approaches. The development of the algorithms allowed not only to visualize the possible combinations in a ternary diagram, but also to predict with good accuracy the formation of new nanoemulsions.

The extension of the model to the prediction of the physicochemical characteristics of nanoparticles represented a further step forward, highlighting the potential of predictive tools as a support for experimentation. This integrated approach paves the way for a more rational and efficient design of drug delivery systems, reducing time and costs and improving the sustainability of research.

6. Future development

The results obtained in this work constitute a solid basis but also open up various development prospects.

First of all, it will be necessary to repeat all the experiments using doxorubicin, so as to verify the consistency of the results compared to the preliminary ones conducted without the drug. In particular, it will be important to confirm critical parameters such as size, polydispersity index and zeta potential, while extending the assessments made previously with Rhodamine B regarding encapsulation efficiency and drug loading to obtain a more complete picture of formulation performance.

Further development will involve the analysis of polymer-drug interactions by spectroscopic and thermoanalytic techniques, in particular FTIR and DSC. FTIR will allow us to identify any changes in the functional groups, revealing possible chemical interactions or specific links between the components, while DSC will allow us to highlight variations in the thermal profiles, providing useful information on the stability of the formulation and the nature of the encapsulation.

It will also be crucial to study the drug release profile at different time scales, to evaluate the ability of nanoparticles to ensure controlled and sustained release. These data will allow us to understand whether the system is able to avoid peaks in systemic concentration, improving therapeutic efficacy and reducing toxicity. Correlation of the release results with the physico-chemical characteristics may finally offer additional optimisation tools.

An equally relevant aspect will be the study of cellular uptake, which is necessary to elucidate the internalization mechanisms of nanoparticles and their intracellular distribution. In parallel, deepening targeting strategies could increase selectivity towards cancer cells, reducing healthy tissue exposure and thus side effects.

To complete the evaluation, it will be essential to perform cytotoxicity tests on cell lines, comparing the efficacy and toxicity of free doxorubicin with that carried by nanoparticles. This will allow us to verify whether encapsulation allows us to reduce side effects without compromising their anti-tumor activity.

Finally, it will be crucial to investigate the transplacental behaviour of nanoparticles, so as to ascertain the possible ability to cross the placental barrier. This test plays a key role in the

context of breast cancer during pregnancy, where foetal safety is a top priority. At the same time, preclinical studies in mouse models will allow the evaluation of pharmacokinetics, biodistribution and therapeutic efficacy in vivo, representing an essential step for future clinical translation.

APPENDIX A: DATASET

WATER %	OIL %	SURFACTANT %	EMULSION(0/1)	SIZE (nm)	PDI	Z POT (mV)	EE %	DL %
70	17.5	12.5	1	835	0.864	-28.63	43.60	10.31
20.0	40.0	40.0	0	0	0	0	0	0
93.5	5	1.5	1	188.18	0.411	-35.43	80.83	11.42
60.0	32.5	7.5	0	0	0	0	0	0
90.0	2.0	8.0	0	0	0	0	0	0
5.0	70.0	25.0	0	0	0	0	0	0
20.0	50.0	30.0	0	0	0	0	0	0
5.0	45.0	50.0	0	0	0	0	0	0
40.0	20.0	40.0	0	0	0	0	0	0
40.0	40.0	20.0	0	0	0	0	0	0
5.0	55.0	40.0	0	0	0	0	0	0
80.0	3.0	17.0	0	0	0	0	0	0
60	39.	0.5	0	0	0	0	0	0
30.0	10.0	60.0	0	0	0	0	0	0
40.0	10.0	50.0	0	0	0	0	0	0
70.0	27.5	2.5	0	0	0	0	0	0
5.0	5.0	90.0	0	0	0	0	0	0
30.0	40.0	30.0	0	0	0	0	0	0
60.0	10.0	30.0	0	0	0	0	0	0
20.0	10.0	70.0	0	0	0	0	0	0
10.0	50.0	40.0	0	0	0	0	0	0
87.5	7.5	5	1	173.09	0.485	-38.13	75.53	11.27
10.0	60.0	30.0	0	0	0	0	0	0
5.0	10.0	85.0	0	0	0	0	0	0
90	5	5	1	848.467	0.689	-30.7	78.06	11.34
5.0	85.0	10.0	0	0	0	0	0	0
10.0	20.0	70.0	0	0	0	0	0	0
80.0	2.0	18.0	0	0	0	0	0	0
50.0	10.0	40.0	0	0	0	0	0	0
87.5	11.5	1	1	417.92	0.198	-25.63	70.72	11.12
70.0	8.0	22.0	0	0	0	0	0	0
92.5	6.5	1	1	395.28	0.369	-31.17	76.19	11.29
85	11	4	1	-1	-1	-1	-1	-1
5.0	20.0	75.0	0	0	0	0	0	0
89	9	2	1	218.03	0.366	-26.3	74.31	11.23
5.0	25.0	70.0	0	0	0	0	0	0
70	15	15	1	1231.32	0.435	-40.67	45.51	10.37
10.0	30.0	60.0	0	0	0	0	0	0
91.2	7.3	1.5	1	268.52	0.371	-35.1	66.66	11.00
90.0	3.0	7.0	0	0	0	0	0	0
85.0	12.5	2.5	0	0	0	0	0	0
90.0	4.0	6.0	0	0	0	0	0	0
95	4	1	1	216.92	0.274	-35.13	74.17	11.23

10.0	40.0	50.0	0	0	0	0	0	0
60.0	37.5	2.5	0	0	0	0	0	0
92.5	7	0.5	1	295.27	0.369	-31.47	77.49	11.32
90	9.5	0.5	1	476.56	0.158	-22.17	48.24	10.45
30.0	50.0	20.0	0	0	0	0	0	0
75	15	10	1	592.1	0.508	-36.57	44.55	10.34
70.0	13.0	17.0	0	0	0	0	0	0
97.5	2.2	0.3	1	287.856	0.335	-29.77	44.40	10.33
70.0	11.0	19.0	0	0	0	0	0	0
30.0	20.0	50.0	0	0	0	0	0	0
90	6	4	1	-1	-1	-1	-1	-1
70.0	10.0	20.0	0	0	0	0	0	0
90	9	1	1	350.13	0.206	-27.43	80.05	11.40
70.0	7.0	23.0	0	0	0	0	0	0
80.0	19.5	0.5	0	0	0	0	0	0
85.0	14.0	1.0	0	0	0	0	0	0
80.0	5.0	15.0	0	0	0	0	0	0
80	11.5	8.5	1	496.53	0.514	-32.27	83.79	11.51
10.0	70.0	20.0	0	0	0	0	0	0
5.0	40.0	55.0	0	0	0	0	0	0
20.0	70.0	10.0	0	0	0	0	0	0
80.0	15.0	5.0	0	0	0	0	0	0
80.0	17.5	2.5	0	0	0	0	0	0
86.5	11	2.5	1	207.4	0.284	-32.7	81.91	11.46
5.0	65.0	30.0	0	0	0	0	0	0
91.2	7.8	1	1	277.78	0.213	-28.87	78.38	11.35
80.0	8.0	12.0	0	0	0	0	0	0
5.0	80.0	15.0	0	0	0	0	0	0
20.0	20.0	60.0	0	0	0	0	0	0
5.0	60.0	35.0	0	0	0	0	0	0
80.0	9.0	11.0	0	0	0	0	0	0
80	10	10	1	759.06	0.770	-36.2	83.19	11.50
50.0	40.0	10.0	0	0	0	0	0	0
88	10.5	1.5	1	259.64	0.179	-32.03	86.10	11.58
70.0	12.0	18.0	0	0	0	0	0	0
90	8	2	1	-1	-1	-1	-1	-1
5.0	35.0	60.0	0	0	0	0	0	0
60.0	35.0	5.0	0	0	0	0	0	0
50.0	30.0	20.0	0	0	0	0	0	0
70.0	6.0	24.0	0	0	0	0	0	0
87.5	10	2.5	1	244.65	0.213	-26.13	87.47	11.62
90.0	0.5	9.5	0	0	0	0	0	0
90	7.5	2.5	1	-1	-1	-1	-1	-1
70.0	4.0	26.0	0	0	0	0	0	0
30.0	30.0	40.0	0	0	0	0	0	0
70.0	29.5	0.5	0	0	0	0	0	0

5.0	50.0	45.0	0	0	0	0	0	0
80.0	1.0	19.0	0	0	0	0	0	0
90	7	3	1	-1	-1	-1	-1	-1
20.0	60.0	20.0	0	0	0	0	0	0
80.0	4.0	16.0	0	0	0	0	0	0
70.0	9.0	21.0	0	0	0	0	0	0
70.0	1.0	29.0	0	0	0	0	0	0
5.0	30.0	65.0	0	0	0	0	0	0
94.5	4	1.5	1	179.39	0.282	-33.83	80.97	11.43
10.0	10.0	80.0	0	0	0	0	0	0
95	4.7	0.3	1	355.87	0.337	-28.57	13.82	9.41
50.0	20.0	30.0	0	0	0	0	0	0
40.0	30.0	30.0	0	0	0	0	0	0
80.0	7.0	13.0	0	0	0	0	0	0
5.0	90.0	5.0	0	0	0	0	0	0
80.0	12.5	7.5	0	0	0	0	0	0
91.2	8.3	0.5	1	373.17	0.281	-30.93	84.99	11.55
85	10	5	1	180.78	0.370	-33.57	79.03	11.37
70.0	25.0	5.0	0	0	0	0	0	0
70.0	2.0	28.0	0	0	0	0	0	0
90	9.3	0.7	1	384.24	0.133	-25.67	83.08	11.49
30.0	60.0	10.0	0	0	0	0	0	0
70.0	5.0	25.0	0	0	0	0	0	0
70.0	22.5	7.5	0	0	0	0	0	0
80.0	6.0	14.0	0	0	0	0	0	0
90.0	1.0	9.0	0	0	0	0	0	0
70.0	20.0	10.0	0	0	0	0	0	0
20.0	30.0	50.0	0	0	0	0	0	0
40.0	50.0	10.0	0	0	0	0	0	0
70.0	3.0	27.0	0	0	0	0	0	0
90	8.5	1.5	1	264.89	0.214	-26.23	82.08	11.46
5.0	75.0	20.0	0	0	0	0	0	0
85	11.5	3.5	1	217.2	0.238	-29.03	76.37	11.29
70.0	14.0	16.0	0	0	0	0	0	0
96	3.5	0.5	1	265.92	0.321	-31.62	50.75	4.27
89	10	1	1	282.53	0.239	-28.33	79.86	11.40
10.0	80.0	10.0	0	0	0	0	0	0
5.0	15.0	80.0	0	0	0	0	0	0

APPENDIX B: Python Code

```
import numpy as np
import matplotlib.pyplot as plt
import ternary
import seaborn as sns

from sklearn.ensemble import RandomForestClassifier,
GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.multioutput import MultiOutputRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split, cross_val_predict
from sklearn.metrics import mean_squared_error, r2_score

# ========== 1. Loading data ==========
data = np.loadtxt(r"C:\Users\vinzg\Desktop\PoliTO\laurea
magistrale\thesis\python code\new code\DATASET.txt")
np.random.shuffle(data)

X_all_raw = data[:, :3] / 100.0
y_class = data[:, 3]
size, pdi, zpot, ee, dl = data[:, 4], data[:, 5], data[:, 6], data[:, 7],
data[:, 8]

# ==== Filter: Water >= 50% ====
mask_water = X_all_raw[:, 0] >= 0.5
X_all = X_all_raw[mask_water]
y_class = y_class[mask_water]
size = size[mask_water]
pdi = pdi[mask_water]
zpot = zpot[mask_water]
ee = ee[mask_water]
dl = dl[mask_water]

# ========== 2. Classification ==========
X_train_cls, X_test_cls, y_train_cls, y_test_cls = train_test_split(
    X_all, y_class, test_size=0.2, random_state=42)

clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train_cls, y_train_cls)
print(f"🎯 RF Accuracy: {clf.score(X_test_cls, y_test_cls):.2f}")

# ========== 3A. Regression SVR ==========
mask_spz = (y_class == 1) & (size != -1) & (pdi != -1) & (zpot != -1)
X_spz_all = X_all[mask_spz]
y_spz_all = np.vstack([size[mask_spz], pdi[mask_spz], zpot[mask_spz]]).T
```

```
X_train_spz, X_test_spz, y_train_spz, y_test_spz = train_test_split(
    X_spz_all, y_spz_all, test_size=0.2, random_state=42)

svr_base = make_pipeline(StandardScaler(), SVR(kernel='rbf', C=100,
epsilon=0.1))
svr_spz = MultiOutputRegressor(svr_base)
svr_spz.fit(X_train_spz, y_train_spz)

# Evaluation + Cross-Validation
y_pred_cv_spz = cross_val_predict(svr_spz, X_spz_all, y_spz_all, cv=5)

print("\n📊 SVR Evaluation (Size, PDI, Zeta Potential):")
for i, name in enumerate(["Size", "PDI", "Zeta Potential"]):
    mse = mean_squared_error(y_spz_all[:, i], y_pred_cv_spz[:, i])
    r2 = r2_score(y_spz_all[:, i], y_pred_cv_spz[:, i])
    print(f" - {name}: MSE = {mse:.2f}, R² = {r2:.2f}")

# Plot
fig, axs = plt.subplots(1, 3, figsize=(15, 4))
for i, name in enumerate(["Size", "PDI", "Zeta Potential"]):
    axs[i].scatter(y_spz_all[:, i], y_pred_cv_spz[:, i], alpha=0.6)
    axs[i].plot([min(y_spz_all[:, i]), max(y_spz_all[:, i])],
                [min(y_spz_all[:, i]), max(y_spz_all[:, i])], 'r--')
    axs[i].set_title(name)
    axs[i].set_xlabel("Actual")
    axs[i].set_ylabel("Predicted")
plt.suptitle("SVR - Regression")
plt.tight_layout()
plt.show()

# ========== 3B. Regression GBR ==========
mask_eddl = (y_class == 1) & (ee != -1) & (dl != -1)
X_eddl_all = X_all[mask_eddl]
y_eddl_all = np.vstack([ee[mask_eddl], dl[mask_eddl]]).T

X_train_eddl, X_test_eddl, y_train_eddl, y_test_eddl = train_test_split(
    X_eddl_all, y_eddl_all, test_size=0.2, random_state=42)

gbr_base = GradientBoostingRegressor(n_estimators=100, random_state=42)
gbr_model = MultiOutputRegressor(gbr_base)
gbr_model.fit(X_train_eddl, y_train_eddl)

y_pred_cv_eddl = cross_val_predict(gbr_model, X_eddl_all, y_eddl_all,
cv=5)

print("\n📊 GBR Evaluation (EE%, DL%):")
for i, name in enumerate(["EE%", "DL%"]):
    mse = mean_squared_error(y_eddl_all[:, i], y_pred_cv_eddl[:, i])
    r2 = r2_score(y_eddl_all[:, i], y_pred_cv_eddl[:, i])
```

```
    print(f" - {name}: MSE = {mse:.2f}, R² = {r2:.2f}")

fig, axs = plt.subplots(1, 2, figsize=(10, 4))
for i, name in enumerate(["EE%", "DL%"]):
    axs[i].scatter(y_eddl_all[:, i], y_pred_cv_eddl[:, i], alpha=0.6)
    axs[i].plot([min(y_eddl_all[:, i]), max(y_eddl_all[:, i])],
                [min(y_eddl_all[:, i]), max(y_eddl_all[:, i])], 'r--')
    axs[i].set_title(name)
    axs[i].set_xlabel("Actual")
    axs[i].set_ylabel("Predicted")
plt.suptitle("Gradient Boosting - Regression")
plt.tight_layout()
plt.show()

# ========== 4. Ternary diagram ==========
pred_class = []
for x in X_all_raw:
    if x[0] < 0.5:
        pred_class.append(0)
    else:
        pred_class.append(clf.predict([x])[0])
pred_class = np.array(pred_class)

emul = X_all_raw[pred_class == 1] * 100
non_emul = X_all_raw[pred_class == 0] * 100

fig, tax = ternary.figure(scale=100)
tax.boundary(linewidth=2.0)
tax.gridlines(multiple=10, color="gray", linewidth=0.5)
tax.set_title("Ternary Diagram", fontsize=18, weight='bold')
tax.left_axis_label("Water %", fontsize=14)
tax.right_axis_label("Surfactant %", fontsize=14)
tax.bottom_axis_label("Oil %", fontsize=14, offset=-0.09)

tax.scatter([(p[1], p[2], p[0]) for p in emul], color='green', marker='o',
label="Emulsion")
tax.scatter([(p[1], p[2], p[0]) for p in non_emul], color='red',
marker='x', label="No Emulsion")
tax.legend()
plt.tight_layout()
plt.show()

# ========== 5. Interactive Prediction ==========
while True:
    print("\n🔢 Enter a triplet (water, oil, surfactant) in percentage.
Sum = 100.")
    try:
        water = float(input(" - Water (%): "))
        oil = float(input(" - Oil (%): "))
        surf = float(input(" - Surfactant (%): "))
```

```
    except ValueError:
        print("❌ Please enter valid numbers.")
        continue

    if round(water + oil + surf, 1) != 100.0:
        print("❌ The sum must be 100%. Try again.")
        continue

    if water < 50:
        print("❌ Water < 50%: No oil-in-water emulsion expected.")
        cont = input("\nDo you want to continue? (y/n): ").strip().lower()
        if cont != 'y':
            break
        else:
            continue

    new_input = np.array([[water, oil, surf]]) / 100.0
    class_pred = clf.predict(new_input)[0]

    if class_pred == 1:
        print("\n✅ Nanoemulsion predicted!")
        size_pred, pdi_pred, zpot_pred = svr_spz.predict(new_input)[0]
        ee_pred, dl_pred = gbr_model.predict(new_input)[0]

        print(f" 📏 Size: {size_pred:.1f} nm")
        print(f" 🔸 PDI: {pdi_pred:.3f}")
        print(f" ⚡ Zeta potential: {zpot_pred:.1f} mV")
        print(f" 🧪 EE%: {ee_pred:.1f}")
        print(f" 💊 DL%: {dl_pred:.1f}")
    else:
        print("\n❌ No nanoemulsion predicted.")

    cont = input("\nDo you want to enter another triplet? (y/n):
").strip().lower()
    if cont != 'y':
        print("Interaction ended.")
        break
```

Bibliography

[1] Akhlaqi, M., Ghofrani, A., Najdi, N., Ranjkesh, M., … & Almasi-Hashiani, A. (2025). *A systematic review and meta-analysis of pregnancy-associated breast cancer incidence rate.* BMC Cancer, **25**, Article 660.

[2] P. Schedin, 'Pregnancy-associated breast cancer and metastasis', *Nat. Rev. Cancer*, vol. 6, no. 4, pp. 281–291, Apr. 2006, doi: 10.1038/nrc1839.

[3] E. M. Elmowafy, M. Tiboni, and M. E. Soliman, 'Biocompatibility, biodegradation and biomedical applications of poly(lactic acid)/poly(lactic-co-glycolic acid) micro and nanoparticles', *J. Pharm. Investig.*, vol. 49, no. 4, pp. 347–380, Jul. 2019, doi: 10.1007/s40005-019-00439-x.

[4] I. Paris *et al.*, 'Pregnancy-Associated Breast Cancer: A Multidisciplinary Approach', *Clin. Breast Cancer*, vol. 21, no. 1, pp. e120–e127, Feb. 2021, doi: 10.1016/j.clbc.2020.07.007.

[5] K. Varagur *et al.*, 'Understanding Trends in Incidence and Management of Pregnancy-Associated Breast Cancer in a National Sample Using Claims Data', *J. Surg. Oncol.*, vol. n/a, no. n/a, doi: 10.1002/jso.70010.

[6] A. Mumtaz, N. Otey, B. Afridi, and H. Khout, 'Breast cancer in pregnancy: a comprehensive review of diagnosis, management, and outcomes', *Transl. Breast Cancer Res.*, vol. 5, p. 21, Jul. 2024, doi: 10.21037/tbcr-24-26.

[7] K. S. Asgeirsson, 'Pregnancy-associated breast cancer', *Acta Obstet. Gynecol. Scand.*, vol. 90, no. 2, pp. 158–166, 2011, doi: 10.1111/j.1600-0412.2010.01035.x.

[8] 'Quaderno_ROPI_Mammario_Gravidanza_09022022.pdf'. Accessed: Jun. 16, 2025. [Online]. Available: https://www.reteoncologicaropi.it/wp-content/uploads/2022/03/Quaderno_ROPI_Mammario_Gravidanza_09022022.pdf

[9] M. Michalska *et al.*, 'Diagnostic and Therapeutic Challenges in Pregnancy-Associated Breast Cancer: A Literature Review', *Qual. Sport*, vol. 42, Jun. 2025, doi: 10.12775/QS.2025.42.60515.

[10] A. L. V. Johansson, C. E. Weibull, I. Fredriksson, and M. Lambe, 'Diagnostic pathways and management in women with pregnancy-associated breast cancer (PABC): no evidence of treatment delays following a first healthcare contact', *Breast Cancer Res. Treat.*, vol. 174, no. 2, pp. 489–503, Apr. 2019, doi: 10.1007/s10549-018-05083-x.

[11] F. Galati *et al.*, 'Pregnancy-Associated Breast Cancer: A Diagnostic and Therapeutic Challenge', *Diagnostics*, vol. 13, no. 4, p. 604, Feb. 2023, doi: 10.3390/diagnostics13040604.

[12] M. P. Mano and D. M. Bortolini, 'CARCINOMA DELLA MAMMELLA ASSOCIATO A GRAVIDANZA (CAMG)'.

[13] 'Treating Breast Cancer During Pregnancy'. Accessed: Jun. 11, 2025. [Online]. Available: https://www.cancer.org/cancer/types/breast-cancer/treatment/treating-breast-cancer-during-pregnancy.html

[14] S. Islam, M. M. S. Ahmed, M. A. Islam, N. Hossain, and M. A. Chowdhury, 'Advances in nanoparticles in targeted drug delivery–A review', *Results Surf. Interfaces*, vol. 19, p. 100529, May 2025, doi: 10.1016/j.rsurfi.2025.100529.

[15] M. A. Beach *et al.*, 'Polymeric Nanoparticles for Drug Delivery', *Chem. Rev.*, vol. 124, no. 9, pp. 5505–5616, May 2024, doi: 10.1021/acs.chemrev.3c00705.

[16] 'The Evolution of Nanomedicine: from ideas to clinical applications', Inside Therapeutics. Accessed: Jun. 17, 2025. [Online]. Available: https://insidetx.com/review/the-evolution-of-nanomedicine-from-ideas-to-clinical-applications/

[17] I. Khan, K. Saeed, and I. Khan, 'Nanoparticles: Properties, applications and toxicities', *Arab. J. Chem.*, vol. 12, no. 7, pp. 908–931, Nov. 2019, doi: 10.1016/j.arabjc.2017.05.011.

[18] 'Nano Based Drug Delivery Systems: Present and Future Prospects'. Accessed: Jun. 17, 2025. [Online]. Available: https://www.scientificliterature.org/Nanomedicine/Nanomedicine-22-121.pdf

[19] D. S. Soliman, 'Nanomedicine: Advantages and Disadvantages of Nanomedicine', *J. Nanomedicine Nanotechnol.*, vol. 14, no. 3, pp. 1–2, Mar. 2023, doi: 10.35248/2157-7439.23.14.666.

[20] M. R. Gwinn and V. Vallyathan, 'Nanoparticles: Health Effects—Pros and Cons', *Environ. Health Perspect.*, vol. 114, no. 12, pp. 1818–1825, Dec. 2006, doi: 10.1289/ehp.8871.

[21] 'Nanotechnology in medicine: Risks and benefits', Held+Team. Accessed: Jun. 17, 2025. [Online]. Available: https://heldundteam.de/en/aktuelles/nanotechnologie-in-der-medizin

[22] S. Tran, P.-J. DeGiovanni, B. Piel, and P. Rai, 'Cancer nanomedicine: a review of recent success in drug delivery', *Clin. Transl. Med.*, vol. 6, no. 1, p. 44, Dec. 2017, doi: 10.1186/s40169-017-0175-0.

[23] D. D. W. Team, 'Nanoparticles & Nanomedicines - Exploring The Past, Present and Future', Drug Discovery World (DDW). Accessed: Jul. 21, 2025. [Online]. Available: https://www.ddw-online.com/nanoparticles-nanomedicines-exploring-the-past-present-and-future-1247-201710/

[24] A. Haleem, M. Javaid, R. P. Singh, S. Rab, and R. Suman, 'Applications of nanotechnology in medical field: a brief review', *Glob. Health J.*, vol. 7, no. 2, pp. 70–77, Jun. 2023, doi: 10.1016/j.glohj.2023.02.008.

[25] C.-Y. Hsu *et al.*, 'An overview of nanoparticles in drug delivery: Properties and applications', *South Afr. J. Chem. Eng.*, vol. 46, pp. 233–270, Oct. 2023, doi: 10.1016/j.sajce.2023.08.009.

[26] 'PLGA | Nanovex'. Accessed: Jun. 18, 2025. [Online]. Available: https://www.nanovexbiotech.com/plga/

[27] S. Sharma, A. Parmar, S. Kori, and R. Sandhir, 'PLGA-based nanoparticles: A new paradigm in biomedical applications', *TrAC Trends Anal. Chem.*, vol. 80, pp. 30–40, Jun. 2016, doi: 10.1016/j.trac.2015.06.014.

[28] H. K. Makadia and S. J. Siegel, ‘Poly Lactic-co-Glycolic Acid (PLGA) as Biodegradable Controlled Drug Delivery Carrier’, *Polymers*, vol. 3, no. 3, pp. 1377–1397, Sep. 2011, doi: 10.3390/polym3031377.

[29] N. V. García, ‘APLICACIONES TERAPÉUTICAS DEL ÁCIDO POLI LÁCTICO-CO-GLICÓLICO (PLGA)’.

[30] H. H. Tayeb and F. Sainsbury, ‘Nanoemulsions in Drug Delivery: Formulation to Medical Application’, *Nanomed.*, vol. 13, no. 19, pp. 2507–2525, Oct. 2018, doi: 10.2217/nnm-2018-0088.

[31] M. Kreilgaard, ‘Influence of microemulsions on cutaneous drug delivery’, *Adv. Drug Deliv. Rev.*, vol. 54, pp. S77–S98, Nov. 2002, doi: 10.1016/S0169-409X(02)00116-3.

[32] K. R. B. Singh, J. Singh, C. O. Adetunji, and R. Pratap Singh, *Nanotechnology for drug delivery and pharmaceuticals*. London, England: Academic Press, 2023.

[33] M. Kumar, R. S. Bishnoi, A. K. Shukla, and C. P. Jain, ‘Techniques for Formulation of Nanoemulsion Drug Delivery System: A Review’, *Prev. Nutr. Food Sci.*, vol. 24, no. 3, pp. 225–234, Sep. 2019, doi: 10.3746/pnf.2019.24.3.225.

[34] ‘Ternary Diagrams’, Ternary Diagrams. Accessed: Jul. 16, 2025. [Online]. Available: https://serc.carleton.edu/mathyouneed/geomajors/ternary/index.html

[35] N. E. Hamilton and M. Ferry, ‘ggtern: Ternary Diagrams Using ggplot2’, *J. Stat. Softw.*, vol. 87, no. 1, pp. 1–17, Dec. 2018, doi: 10.18637/jss.v087.c03.

[36] D. R. F. West, *Ternary Phase Diagrams in Materials Science*. Milton, UNITED KINGDOM: Taylor & Francis Group, 2002. Accessed: Jul. 16, 2025. [Online]. Available: http://ebookcentral.proquest.com/lib/polito-ebooks/detail.action?docID=3016975

[37] ‘Diagramma ternario’. Accessed: Jul. 16, 2025. [Online]. Available: https://chimicamo.org/chimica-fisica/diagramma-ternario/

[38] N. E. Hamilton and M. Ferry, ‘ggtern: Ternary Diagrams Using ggplot2’, *J. Stat. Softw.*, vol. 87, pp. 1–17, Dec. 2018, doi: 10.18637/jss.v087.c03.

[39] S. A. Bini, ‘Artificial Intelligence, Machine Learning, Deep Learning, and Cognitive Computing: What Do These Terms Mean and How Will They Impact Health Care?’, *J. Arthroplasty*, vol. 33, no. 8, pp. 2358–2361, Aug. 2018, doi: 10.1016/j.arth.2018.02.067.

[40] K. Sharifani and M. Amini, ‘Machine Learning and Deep Learning: A Review of Methods and Applications’, vol. 10, no. 07, 2023.

[41] C. Janiesch, P. Zschech, and K. Heinrich, ‘Machine learning and deep learning’, *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.

[42] J. G. Greener, S. M. Kandathil, L. Moffat, and D. T. Jones, ‘A guide to machine learning for biologists’, *Nat. Rev. Mol. Cell Biol.*, vol. 23, no. 1, pp. 40–55, Jan. 2022, doi: 10.1038/s41580-021-00407-0.

[43] A. B. Çolak and S. A. Onaizi, ‘Machine learning approach for modelling and predicting interfacial tension and rheology of crude oil nanoemulsions stabilized by rhamnolipid biosurfactant’, *Pet. Res.*, p. S2096249525000109, Feb. 2025, doi: 10.1016/j.ptlrs.2025.02.005.

[44] S. Yakoubi, I. Kobayashi, K. Uemura, M. Nakajima, I. Hiroko, and M. A. Neves, ‘Recent advances in delivery systems optimization using machine learning approaches’, *Chem. Eng. Process. - Process Intensif.*, vol. 188, p. 109352, Jun. 2023, doi: 10.1016/j.cep.2023.109352.

[45] ‘Cosa è l’apprendimento supervisionato? | IBM’. Accessed: Jun. 30, 2025. [Online]. Available: https://www.ibm.com/it-it/think/topics/supervised-learning

[46] ‘UnSupervised learning, cos’è ed esempi di apprendimento non supervisionato’, AI4Business. Accessed: Jun. 30, 2025. [Online]. Available: https://www.ai4business.it/intelligenza-artificiale/unsupervised-learning-cose-ed-esempi-di-apprendimento-non-supervisionato/

[47] ‘Confronto: apprendimento supervisionato vs apprendimento per rinforzo’. Accessed: Jun. 30, 2025. [Online]. Available: https://iartificial.blog/it/aprendizaje/comparativa-aprendizaje-supervisado-vs-aprendizaje-por-refuerzo/

[48] ‘What is the k-nearest neighbors algorithm? | IBM’. Accessed: Jun. 30, 2025. [Online]. Available: https://www.ibm.com/think/topics/knn

[49] ‘K-Nearest Neighbor (KNN) Explained | Pinecone’. Accessed: Jun. 30, 2025. [Online]. Available: https://www.pinecone.io/learn/k-nearest-neighbor/

[50] ‘KNN for classification and regression’. Accessed: Jun. 30, 2025. [Online]. Available: https://cgi.luddy.indiana.edu/~yye/b565/knn.php

[51] ‘Algoritmo k-Nearest Neighbor | Aprende Machine Learning’. Accessed: Jun. 30, 2025. [Online]. Available: https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/

[52] ‘K-Nearest Neighbor(KNN) Algorithm’, GeeksforGeeks. Accessed: Jul. 24, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/k-nearest-neighbours/

[53] ‘What Is Support Vector Machine? | IBM’. Accessed: Jun. 30, 2025. [Online]. Available: https://www.ibm.com/think/topics/support-vector-machine

[54] ‘Support Vector Regressor’. Accessed: Jun. 30, 2025. [Online]. Available: https://apmonitor.com/pds/index.php/Main/SupportVectorRegressor

[55] ‘Support Vector Regression: A Comprehensive Guide for Machine Learning Practitioners | 33rd Square’. Accessed: Jun. 30, 2025. [Online]. Available: https://33rdsquare.com/tech/ai/support-vector-regression-tutorial-for-machine-learning/

[56] ‘Support Vector Regression - an overview | ScienceDirect Topics’. Accessed: Jun. 30, 2025. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/support-vector-regression

[57] H. Muthiah, U. Sa'adah, and A. Efendi, 'Support Vector Regression (SVR) Model for Seasonal Time Series Data', 2021.

[58] 'Gradient Boosting - an overview | ScienceDirect Topics'. Accessed: Jun. 30, 2025. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/gradient-boosting

[59] 'What is Gradient Boosting? | IBM'. Accessed: Jun. 30, 2025. [Online]. Available: https://www.ibm.com/think/topics/gradient-boosting

[60] 'A Guide to The Gradient Boosting Algorithm'. Accessed: Jun. 30, 2025. [Online]. Available: https://www.datacamp.com/tutorial/guide-to-the-gradient-boosting-algorithm

[61] 'Gradient Boosting in ML', GeeksforGeeks. Accessed: Jul. 24, 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/ml-gradient-boosting/

[62] A. Saxena, 'Polynomial Regression in Machine Learning', Applied AI Blog. Accessed: Jun. 30, 2025. [Online]. Available: https://www.appliedaicourse.com/blog/polynomial-regression-in-machine-learning/

[63] 'Polynomial Regression: An Introduction', Built In. Accessed: Jun. 30, 2025. [Online]. Available: https://builtin.com/machine-learning/polynomial-regression

[64] A. Pe˘ckov, 'A MACHINE LEARNING APPROACH TO POLYNOMIAL REGRESSION'.

[65] R. Agrawal, 'Polynomial Regression for Beginners', Analytics Vidhya. Accessed: Jul. 24, 2025. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/07/all-you-need-to-know-about-polynomial-regression/

[66] 'What Is Random Forest? | IBM'. Accessed: Jul. 01, 2025. [Online]. Available: https://www.ibm.com/think/topics/random-forest

[67] 'Random Forest - an overview | ScienceDirect Topics'. Accessed: Jul. 01, 2025. [Online]. Available: https://www.sciencedirect.com/topics/engineering/random-forest

[68] J. Singh, 'Random Forest: Pros and Cons', Medium. Accessed: Jul. 01, 2025. [Online]. Available: https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04

[69] 'A Comprehensive Guide to K-Fold Cross Validation'. Accessed: Jul. 21, 2025. [Online]. Available: https://www.datacamp.com/tutorial/k-fold-cross-validation?utm_source=chatgpt.com

[70] T. Fushiki, 'Estimation of prediction error by using K-fold cross-validation', *Stat. Comput.*, vol. 21, no. 2, pp. 137–146, Apr. 2011, doi: 10.1007/s11222-009-9153-8.

[71] Y. Jung, 'Multiple predicting K-fold cross-validation for model selection', *J. Nonparametric Stat.*, vol. 30, no. 1, pp. 197–215, Jan. 2018, doi: 10.1080/10485252.2017.1404598.

[72] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella, 'The "K" in K-fold Cross Validation', *Comput. Intell.*, 2012.

[73] P. Refaeilzadeh, L. Tang, and H. Liu, 'Cross-Validation', in *Encyclopedia of Database Systems*, Springer, Boston, MA, 2009, pp. 532–538. doi: 10.1007/978-0-387-39940-9_565.

[74] ‘The principles of dynamic light scattering | Anton Paar Wiki’, Anton Paar. Accessed: Sep. 04, 2025. [Online]. Available: https://wiki.anton-paar.com/ae-en/the-principles-of-dynamic-light-scattering/

[75] A. H. Silva, F. B. Filippin-Monteiro, B. Mattei, B. G. Zanetti-Ramos, and T. B. Creczynski-Pasa, ‘In vitro biocompatibility of solid lipid nanoparticles’, *Sci. Total Environ.*, vol. 432, pp. 382–388, Aug. 2012, doi: 10.1016/j.scitotenv.2012.06.018.