Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Nanoparticle Properties Modeling Using Linear and Logistic Regression Model on Secondary Experiential Data

Dr. Ashok Mhaske[1], Dr. Ambadas Deshmukh[2], Smt. Shilpa Todmal[3], [4]Mr. Amit Nalavde
[1,4]Department of Mathematics, Dada Patil Mahavidhyalaya, Karjat, Dist-Ahilyanagar, India.
[2] Vidyalankar Institute of Technology Mumbai, India.
[3]Department of Computer Science, Dada Patil College Karjat, Dist-Ahilyanagar, India
mhaske.math@gmail.com, ,amitnalavde97@gmail.com

ARTICLE INFO

Received: 05 Nov 2024
Revised: 18 Dec 2024
Accepted: 26 Dec 2024

ABSTRACT

The techniques used to characterize nanoparticles are important to the development of nanotechnology and materials science. Nevertheless, experimental optimization of a nanoparticle properties, including size, stability, and functionality, can be both inefficient and costly. In this study, a predictive framework, based on linear and logistic regressions is constructed, to derive predictive models to assess nanoparticle characteristics employing secondary experimental data sourced from literature and open-access repositories. In this case, linear regression is used to predict size-dependent continuous outcomes, while logistic regression is applied to predict categorical outcomes, like stability. The models are trained on a portion of the dataset which is then held out for validation, and the models' performance is evaluated on metrics including mean squared error, R^2 score, accuracy, and confusion matrices. The regression models built achieved a good degree of accuracy, which demonstrates modeling based on regression can provide reliable outcomes which will save a chemist significant time in experimental optimal synthesis parameter trials. This demonstrates the value of cross-disciplinary intelligent modeling in nanoscience to improve the decision-making process to speed up the development of multifunctional nanoparticles.

Keywords: Nanotechnology, Nanoparticles, Regression, Logistic, Modeling, Python, Machine Learning

1. Introduction

Nanoparticles are dynamic materials in today's science due to their unique characteristics such as nanoscale size, particle size, dimensions, and exceptional surface area-to-volume ratio. A multitude of emerging and existing technologies in medicine, diagnostics, energy, and the environment are focused on these invaluable nanoscale building blocks as of now. Despite the PhD nanoscale science and engineers of materials setting the standard in the performance of such building blocks, the controlled synthesis factors of constituent reactants, temperature, and reaction time can be costly, labor intensive, and time consuming.

As of late, machine learning and other data analytic tools have gained momentum for their success in predicting the properties of certain materials and shaping parameters for other materials. For these, inter and intra material relations such as concentration, temperature, and time in control of a reaction to achieve a granular particle, or the capturing of nanoparticles-dispersed fluid into a phase or the drying of a liquid, can and have become a target for various models of regression, be it linear or logistic. Out of these, straightforward outcomes such as particle size and yield are suited for the former, in contrast to phase capture and stability which are aligned to the later.

Using widely available publications, as well as internal and external, unpublished research data, constructed experimental databases. Relative importance of grain boundaries and size effects in

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

thermal conductivity of nanocrystalline materials introduced by ([Dong, Wen, & Melnik, 2014). (Albanese, Tang, & Chan, 2012) studied and gives the result on the effect of nanoparticle size, shape, and surface chemistry on biological systems. Nanotoxicology: exposure, mechanism, and effects on human health. in new frontiers in environmental toxicology introduced by (John, Wadhwa, & Mathur, 2022). (James, Witten, Hastie, & Tibshirani, 2021) gives the finding and new approach based on linear regression which involve an introduction to statistical learning: with applications in R.

2. Basic Definition

2.1 Nanoparticles: The high surface area to volume ratio makes nanoparticles distinct from bulk materials and lets them exhibit unique chemical, physical, and biological properties compared to bulk materials. They are particles, each with at least one dimension in the range of 1 to 100 nanometers.
2.2 Synthesis Parameters: The experimental conditions which include temperature, concentration of each individual reactant, pH, and the time reaction takes are classed as synthesis parameters and are critical to the formation, size, and stability of the nanoparticles.

2.3 Particle Size: These individual nanoparticles, which are each 1 nanometer in diameter, possess critical size dimensions. This size dimension is known as particle size. Reactivity and stability are major aspects which particle size defines, as well as performance in use.
2.4 Nanoparticle Stability: Nanoparticle stability is the ability to resist the actions of aggregation, chemical degradation, and sedimentation for a given period of time, and under specified conditions. This remains as an untapped area of study, and constant research is required to prove mechanisms of acquiring resistance are present.
2.5 Secondary Data: By definition, secondary data is the information derived from published records, experimental data, and open-access databases made accessible for the community, rather than information from new experimental data.

2.6 Linear regression: Syntheses parameters within which the constrictions of the particle size are placed remains as 1 and 100 nanometers. From a set of 1 or more independent variables, it is possible to determine 1 dependent variable, in this case particle size, having the ability to apply a constriction to measure. This method is known as linear regression.
2.7 Logistic Regression: Logistic Regression is strategically configuring (within a one-to-one function) the probability of a binary outcome variable (like the stabilization of a nanoparticle which is either stable or unstable) and one or more variables that predicts them, using the sigmoid function.
2.8 Mean Squared Error (MSE): MSE is a metric that quantifies how much a regression model 'misses the mark' or lacks precision. It is the average of the squared variations of a given amount and the amount expected of a continuous variable.
2.9 R^2 Score (Coefficient of Determination): The R2 score calculates the amount of variation that a dependent variable has which is 'explained' by the independent variables within a regression model. Conversely, it's a measure of how well a regression model 'fits' or predicts the outcome.
2.10 Confusion Matrix: A confusion matrix is a chart that analyzes the performance of a classification model with respect to true positives, false positives, false negatives, and true negatives.

3. Methodology

3.1 Data Collection: From published literature, freely available databases of nanoparticles, and experimental data sets, secondary data collection was performed. The dataset features synthesis parameters and the corresponding nanoparticle properties as targets which are specified below:
Independent variables: Temperature (°C), pH, Reaction time (minutes), and Reactant Concentration (M).
3.2 Dependent variables: Particle Size (nm) - Linear Regression and Stability (0 - unstable, 1 - stable) - Logistic Regression

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3.3 Data Preprocessing: For the following parameters, missing values are defined as Imputing the mean or median: Data normalization: standardization of zero mean and unit variance is optional. Data splitting: training and test sets (relevant for larger sets).

3.4 Linear Regression Model: Linear regression associates the particle size with the synthesis parameters like:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + ---+ \beta_n X_n + e$$

The value of coefficient β_i calculated using the following normal equation

$$\beta = (X^T X)^{-1}(X^T X)$$

3.5 Logistic Regression Model: Logistic regression is used to predict binary outcomes:

$$P(y = 1|x) = \frac{1}{(1+e^{-z})}$$

where $z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + ---+ \beta_n X_n + e$

5. Machine Learning Approach: Model is generated using python programming. Pandas for data handling. Scikit -learn for regression and evaluation. Seaborn and Matplotlib libraries for visualization.

4. Numerical Example

For 5 simulated experiments:

Table 4.1: Secondary data for linear and logistic regression for training model

Temp	pH	Time	Conc	Particle Size
50.00	7.00	30.00	0.10	20.00
60.00	6.00	45.00	0.20	25.00
55.00	7.50	40.00	0.15	22.00
65.00	6.50	50.00	0.25	28.00
52.00	7.00	35.00	0.12	21.00

Linear regression model:

Consider the linear equation

$$y = \beta_0 + \beta_1 * Temp + \beta_2 * PH + \beta_3 * Time + \beta_4 * conc$$

Assuming coefficients: $\beta = [-5.00, 0.30, 0.50, 0.20, 10.0]$

Predicted particle size for Experiment 3:

$$\hat{y} = -5 + (0.3 \cdot 55) + (0.5 \cdot 7.5) + (0.2 \cdot 40) + (10 \cdot 0.15)$$

Simplify $\hat{y} = -24.75$ nm

Logistic Regression:

Assuming coefficients: $\beta = [-10.00, 0.20, 1.00, 0.10, 50.0]$

Predicted probability of stability for Experiment 2:

$$z = -10.00 + (0.20 * 60) + (1.00 * 60.00) + (0.10 * 45.00) + (50.00 * 0.20) = 22.50$$

$$P(y = 1) = \frac{1}{1 + e^{-22.50}} \approx 0.999 \approx 1 \Rightarrow Stable$$

5. Python Program: Python program to Modeling Using Linear and Logistic Regression Model on Secondary Experiential Data

```
import pandas as pd  # Included pandas library for analysis of data
```

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
import numpy as np  # Numpy library for numerical calculation
from sklearn.linear_model import LinearRegression, LogisticRegression # sklearn library
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score,
confusion_matrix
import matplotlib.pyplot as plt       # For visualization of matplotlib
import seaborn as sns                 # seaborn library for visualization
data = {
   'Temp': [50, 60, 55, 65, 52],  #list values for Temperature
   'pH': [7, 6, 7.5, 6.5, 7],    #list values for PH
   'Time': [30, 45, 40, 50, 35],  #list values for Time
   'Concen': [0.1, 0.2, 0.15, 0.25, 0.12], #list values for Concentration
   'Part': [20, 25, 22, 28, 21], #list values for particle size
   'Stab': [1, 0, 1, 0, 1] #list values for stability
}
df = pd.DataFrame(data)  # dataframe for tabular form
X = df[['Temp','pH','Time','Concen']]  # created dataframe
y_size = df['Part']    # created dataframe for particle size
y_stability = df['Stab'] # created dataframe stability
lin_model = LinearRegression() # created object of linear regression
lin_model.fit(X, y_size)  # trained data
df['PredictedSize'] = lin_model.predict(X) # Prediction for accuracy
log_model = LogisticRegression() # Created object for logistic regression
log_model.fit(X, y_stability) # Trained the given data
df['PredictedStability'] = log_model.predict(X) # prediction for new data
df['StabilityProb'] = log_model.predict_proba(X)[:,1] # Calculated probability
print("Linear Regression R2:", r2_score(y_size, df['PredictedSize']))
print("Logistic Regression Accuracy:", accuracy_score(y_stability, df['PredictedStability']))
print("Confusion Matrix:\n", confusion_matrix(y_stab, df['PredictedStability'])) # Matrix
plt.figure(figsize=(6,5))
sns.scatterplot(x='ParticleSize', y='PredictedSize', data=df, s=100, color='blue') # sactterplot
plt.plot([min(df['ParticleSize']), max(df['ParticleSize'])],
      [min(df['ParticleSize']), max(df['ParticleSize'])],
      color='red', linestyle='--', label='Ideal Prediction')
plt.xlabel("Actual Particle Size (nm)") # x-axis label
plt.ylabel("Predicted Particle Size (nm)") # y-axis label
plt.title("Linear Regression: Actual vs Predicted Particle Size") # Title of the graph
plt.figure(figsize=(6,5))
sns.barplot(x=df.index, y='StabilityProb', data=df, palette='viridis')
plt.scatter(df.index, df['Stability'], color='red', marker='o', label='Actual Stability')
plt.xlabel("Experiment Index")
plt.ylabel("Predicted Stability Probability")
plt.title("Logistic Regression: Predicted Stability Probabilities")
plt.legend()
plt.show()
plt.figure(figsize=(6,5))
sns.heatmap(df[['Temperature','pH','Time','Concentration','ParticleSize']].corr(),
        annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
from sklearn.metrics import roc_curve, auc
y_true = y_stability #
```

Journal of Information Systems Engineering and Management
2024, 9(4s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
y_scores = df['StabilityProb']  # predicted probabilities from logistic regression
fpr, tpr, thresholds = roc_curve(y_true, y_scores)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(6,5))# figure size
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (AUC = %0.2f)' % roc_auc)
plt.plot([0,1], [0,1], color='red', linestyle='--')  # red line for data prediction
plt.xlim([0.0,1.0])  #x-axis range
plt.ylim([0.0,1.05]) #y-axis range
plt.xlabel('False Positive Rate') #X-axis label
plt.ylabel('True Positive Rate') # Y axis label
plt.title('Logistic Regression: ROC Curve') # title of the figure
plt.legend(loc="lower right") # legend command for description
plt.show()  # To show the plot
```

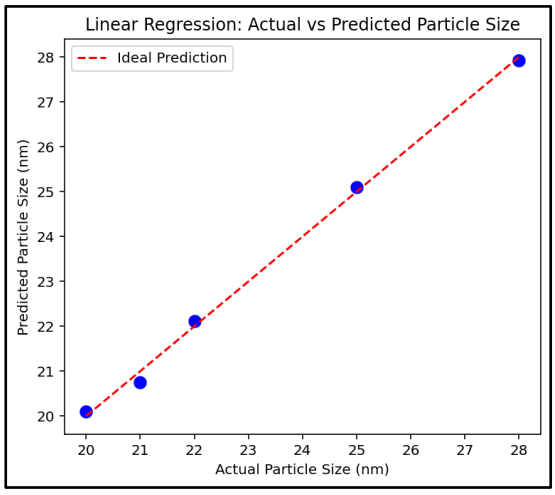

Fig. 5.1 Linear Regression Curve

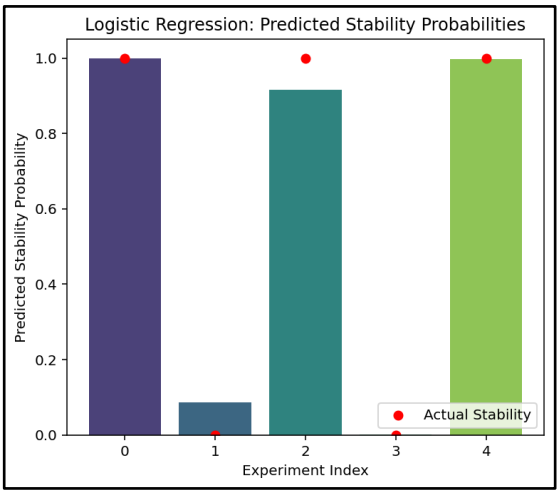

Fig. 5.2 Linear Regression predicted stability probabilities

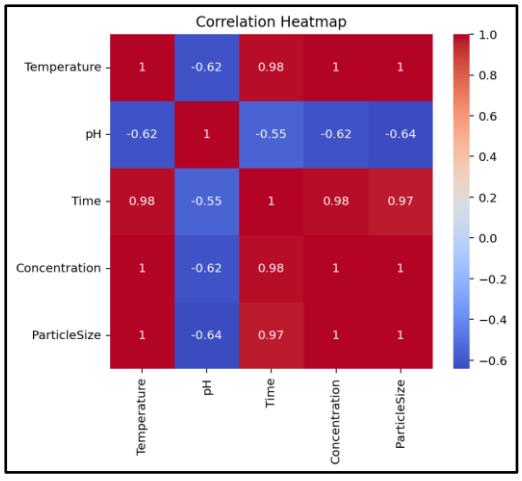

Fig. 5.3 Correlation Hitmap

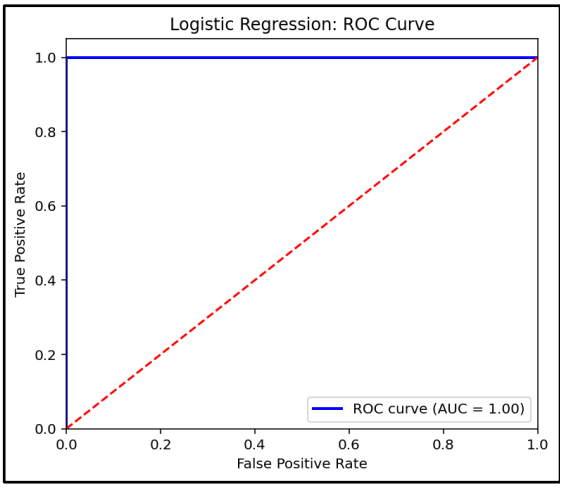

Fig. 5.1 Logistic Regression Curve

6. Conclusion

This study shows that linear and logistic regression can effectively be applied to predict different properties of nanoparticles based on secondary experimental data. This approach is useful for chemists in making informed decisions regarding data integration and minimizing empirical trials.

Linear regression captured the effect of the synthesis parameters on the size of the particles while logistic regression classified the stability of the nanoparticles. Chemists and researchers have a faster way to approach the development of nanoparticles by using regression analysis in nanotechnology research. It also works to developing economic policies on nanotechnology.

7.References

[1]. Dong, H.; Wen, B.; Melnik, R. Relative Importance of Grain Boundaries and Size Effects in Thermal Conductivity of Nanocrystalline Materials. Sci. Rep. 2014, 4 (1), 7037.
[2]. Albanese, A.; Tang, P. S.; Chan, W. C. W. The Effect of Nanoparticle Size, Shape, and Surface Chemistry on Biological Systems. Annu. Rev. Biomed. Eng. 2012, 14, 1–16.
[3]. John, A. T.; Wadhwa, S.; Mathur, A. Nanotoxicology: Exposure, Mechanism, and Effects on Human Health. In New Frontiers in Environmental Toxicology; Jindal, T., Ed.; Springer International Publishing: Cham, 2022; 35–77.
[4]. Pharmacokinetic (PBPK) Modeling of Nanomaterials. ACS Pharmacol. Transl. Sci. 2024, 7 (8), 2251–2279.
[5]. Chou, W.-C.; Chen, Q.; Yuan, L.; Cheng, Y.-H.; He, C.; Monteiro-Riviere, N. A.; Riviere, J. E.; Lin, Z. An Artificial Intelligence-Assisted Physiologically-Based Pharmacokinetic Model to Predict Nanoparticle Delivery to Tumors in Mice. J. Controlled Release 2023, 361, 53–63.
[6]. Kasyanova, V. V.; Bazhukova, I. N. Modeling of Cerium Oxide Nanoparticles Pharmacokinetics. AIP Conf. Proc. 2020, 2313 (1), 080015.
[7]. Blanco, E.; Shen, H.; Ferrari, M. Principles of Nanoparticle Design for Overcoming Biological Barriers to Drug Delivery. Nat. Biotechnol. 2015, 33 (9), 941–951.
[8]. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. Linear Regression. In An Introduction to Statistical Learning: with Applications in R; James, G., Witten, D., Hastie, T., Tibshirani, R., Eds.; Springer US: New York, NY, 2021; pp 59–128.
[9]. Raudys, S. J.; Jain, A. K. Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners. IEEE Trans. Pattern Anal. Mach. Intell. 1991, 13 (3), 252–264.
[10]. Kumar, M.; Kulkarni, P.; Liu, S.; Chemuturi, N.; Shah, D. K. Nanoparticle Biodistribution Coefficients: A Quantitative Approach for Understanding the Tissue Distribution of Nanoparticles. Adv. Drug Delivery Rev. 2023, 194, 114708.
[11]. Soetaert, K.; Petzoldt, T. Inverse Modelling, Sensitivity and Monte Carlo Analysis in R Using Package FME. J. Stat. Softw. 2010, 33, 1–28.
[12]. Moré, J. J. The Levenberg-Marquardt Algorithm: Implementation and Theory. In Numerical Analysis; Watson, G. A., Ed.; Springer: Berlin, Heidelberg, 1978; pp 105–116.
[13]. Marin, J.-M.; Robert, P.. In Bayesian Core: A Practical Approach to Computational Bayesian Statistics; Springer Texts in Statistics; Springer: New York, NY, 2007.