PROTEIN EXTRACTION FROM OILSEEDS: AI-BASED PROCESS OPTIMIZATION

Task submitted for the Completion of course name (ICTH 202)

BACHELOR OF TECHNOLOGY

In

Food Science and Technology

By

Anshika Singh (12203874)

Nandini Mudgil (12213612)

Simarpreet Kaur (12200234)

Gurpreet Kaur (12211667)

Mohammad Sahil Chippa (12201988)

Supervised By

Dr. Piyush Kashyap (27419)

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB

April 2025

DECLARATION

We hereby declare that the work done and presented in the Project/Survey/Task entitled **“protein extraction from oilseeds: ai-based process optimization”** is our own and original. The work has been carried out by us at the School of Agriculture, Lovely Professional University, Phagwara, and Punjab, India under the guidance of Dr. Piyush Kashyap, Assistant Professor (Food Technology and Nutrition) of the School of Agriculture, Lovely Professional University.

Date: 15/04/2025

Place: Phagwara, Punjab (India)

I certify that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 15/04/25	**Supervisor- Dr. Piyush Kashyap (27419)**
Place: Phagwara, Punjab (India)	Food Technology and Nutrition

<u>CERTIFICATE</u>

This is to certify that work embodied in the Task/Survey report entitled **"protein extraction from oilseeds: ai-based process optimization"** has been carried out by under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigations and study. No part of the project has ever been submitted for any other degree. The work has been carried out by her at the School of Agriculture, Lovely Professional University, Phagwara, and Punjab, India. They fulfilled the requirement for the course **ICTH 202.**

Date: 15/04/25

Dr. Piyush Kashyap

Assistant Professor

Department of Food Technology and Nutrition,

School of Agriculture

Lovely Professional University

Phagwara, Punjab, India.

TABLE OF CONTENT

S. No.	Chapter Number	Title	Page Number
1.	Chapter-1	Introduction	5-6
2.	Chapter-2	Review of Literature	7-8
3.	Chapter-3	Problem statement and Solution	9-14
4.	Chapter-4	Result and discussion	15
5.	Chapter-5	Future Perspective and Conclusion	16-18
		References	19

Chapter-1

INTRODUCTION

Machine learning has come so far, it has left a mark in the fields of health, business, and even hobbies. We now try to find out new ways to predict the concentration of protein in samples. It is necessary that the content of protein in food is accurately forecasted for the purpose of not only quality control in food processing, but also the planning of a biotechnological process to be optimized. We aim to create a reliable, efficient, and easily interpretable predictive tool for protein quantification, and the research has been inspired by this goal. The different regression models utilized in the study are Linear Regression, Decision Tree Regressor, Random Forest Regressor, and XGBoost Regressor. All of these models are equipped to excel in a different way, although they share the capability of executing the nonlinear mode and capturing complex interactions between input variables. The dataset was taken from an Excel file, which provided information about the features like pH, temperature, and processing time that could be used to predict the protein content percentage. These models' effectiveness was judged with the help of different statistical indicators such as Mean Squared Error (MSE) and the Coefficient of Determination (R^2). The use of these metrics made it possible to assess the accuracy and stability of the model in predicting protein levels. The GUI in Python has been coded as a means to accompany the work on predictive models. The GUI has a two-fold purpose , as it enables one to interactively compare the results generated by predictive models and also to provide an intuitive visual illustration of the outputs of different models. The main parts of the interface number three. The section on the top includes a list of the performance indicators for all the models, thus instantly indicating the performance level of each model. The middle section is the one that focuses on input and forecast, where the users can type in the values for the relevant features and get the corresponding protein content predictions. The last section is a live bar chart that visually compares the computed protein content of each model. The integration of computation and visualization has the intention to develop the predictive tool in a way that it is not only accessible but informative for the potential end-users. This study aims to determine the most appropriate regression model to estimate protein content and thus makes an important contribution to applied machine learning in the life sciences. The project also has as an aim the creation of a model that is understandable and, therefore, feasible for users. By experimenting with a range of algorithms on a shared platform, this study not only achieves that goal but also proposes the strengths and weaknesses of each model and identifies ensemble techniques as a method of improving prediction accuracy. Under such an umbrella, the selection of a model is informed by quantitative measurements through error metrics as well as qualitative decisions enabled by an open platform. Besides, the study provides a framework for future study into the integration of such prediction tools into more integrated analytical tools in the biotechnology and food science industries. By creating an open demonstration of model utility and effectiveness, this work aims to bridge the gap between theoretical developments in machine learning and applied implementation in production environments. The implications of these findings may enable more streamlined quality control processes, which in turn may be cost-effective and lead to greater product consistency. In summary, this research represents an exhaustive approach to predictive analytics in the context of protein quantification. It combines sophisticated machine learning methodologies with stringent evaluation strategies and skilled visualization tools. The implications of this research are expected to benefit scholars and professionals seeking to apply data-driven methods for quality assessment and process improvement.

Chapter-2

REVIEW OF LITERATURE

A Systematic Analysis of Regression Models for Protein Engineering

The paper consists of systematic approach the obtain estimated set of data for protein analysis through regression and analyse the model on factors such as: presentation and distribution of data, observation distribution and the method of conducting the analysis. In the research, the data was analysed through supervised learning where, the data sets are pre-determined and the regression model is obtained through wildtype. The constituents of the protein were pre-determined of the regression pipeline to particularly test the assessment stage. The data obtained was dramatically different due to differences in the data selection and design train and test splits were designed in a manner to obtain the generalized capability of the task.

Establishment of General Predicted Model for Protein Content in Various Varieties and Colors of Peas using Visible Near- Infrared- Spectroscopy

The study was conducted to estimate the germplasm and determine the protein content of the peas. Spectroscopy at different wavelengths including 400~982nm, 982-~2520nm, and 400~2520nm was used. The protein detection method of PLSR (Partial Least Square Regression) was used. The best range was 400~982nm. The results obtain showed the correlation coefficients between laboratory and predicted values were 0.938 and 0.952 for whole seed and ground seed. In this research the experiment was performed using visible infrared spectroscopy to explore a general-purpose predictions for different varieties of peas.

Benchmarking Uncertainty Quantification for Protein Engineering

In this research, the author tried to create an ML model with better protein design that, estimates accurate combination and determine its accuracy and confidence level of the prediction. Use of Bayesian optimization or active learning model to determine the accurate uncertainty model, to make better choices. Here, they used FLIP (Fitness Landscape Inference for Protein) it is a model that determines the fitness of protein based on its sequence. Also, compared each dataset with various conditions to determine the effectiveness of the model. They concluded that no single uncertainty model was best in either case as, some top predictions worked effectively rather than choosing uncertainty models.

BayeStab: Predicting Effects of Mutation on Protein Stability with Uncertainty Quantification

In this research, the author used BNN (Bayesian Neural Network) and GNN (Graph Neural Network) combination to predict protein mutation through a deep learning tool ByeStab. This also explains how confident the model is with its predictions that is, uncertainty quantification. They obtained a high accuracy in their data with a strong correlation between predicted and obtained value and it also worked well for new protein variants. They also, predicted that data uncertainty was a major concern of causing error while, the model provided better and accurate results.

Uncertainty Analysis in Protein Disorder Prediction

The author discussed about Intrinsically Disordered Proteins (IDP) and its reliability. IDP's do not have a fixed 3D structure yet they play a major role in message transfer across the cells. There are various tools available for predicting the disorder in protein model but none are effective enough to provide accuracy. The researchers trained multiple prediction models for disorder. Then they created meta-predictors models that combine the predictions of many models (like taking a vote or average). These meta-predictors help handle uncertainty better. Meta predictors are ML model that learns through output of the model rather than reading the original data. Their model provided better and accurate results as compared to any other model- it was sensitive and specific. They also observed that their model mutated disordered structure into an ordered form while the reverse was rare. Also, disordered region was fragile as compared to ordered one. Hence, it also concludes that disordered regions were more flexible, evolved faster and are less conserved that structured ones.

Chapter-3

PROBLEM STATEMENT AND SOLUTION

PROBLEM STATEMENT

Protein extraction from oilseeds is a critical step in producing high-quality plant-based proteins used in food, pharmaceutical, and cosmetic industries. However, the extraction process is highly sensitive to multiple parameters such as temperature, pH, solvent type, extraction time, and solid-to-solvent ratio. Conventional methods for optimizing these parameters are often time-consuming, resource-intensive, and fail to achieve maximum efficiency and consistency across different seed types.

To address this challenge, there is a growing need for a data-driven, intelligent approach that can model complex relationships between input variables and output yield. Artificial Intelligence (AI) offers a promising solution by enabling rapid, accurate predictions and optimization of process parameters. This project aims to implement an AI-based system to analyse existing data, predict optimal conditions, and enhance the efficiency of protein extraction from oilseeds, ultimately reducing experimentation time and improving overall yield and purity.

SOLUTION

```
import tkinter as tk
from tkinter import ttk, messagebox
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, r2_score
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
# Load dataset
mydata = pd.read_excel("C:/Users/hp/Downloads/protein.xlsx")
# Split dataset into features () and target (y)
x = mydata.drop('Protein content (%)', axis=1)
y = mydata['Protein content (%)']
x = np.array(x)
y = np.array(y)
# Train-test split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
# Train models
linear_model = LinearRegression().fit(x_train, y_train)
dec_model = DecisionTreeRegressor().fit(x_train, y_train)
random_model = RandomForestRegressor().fit(x_train, y_train)
xgb_model = XGBRegressor().fit(x_train, y_train)
# Evaluate models
models = {
    "Linear Regression": linear_model,
    "Decision Tree": dec_model,
    "Random Forest": random_model,
    "XGBoost": xgb_model
}
results = []
for name, model in models.items():
    y_train_pred = model.predict(x_train)
    y_test_pred = model.predict(x_test)
    mse_train = mean_squared_error(y_train, y_train_pred)
    mse_test = mean_squared_error(y_test, y_test_pred)
    r2_train = r2_score(y_train, y_train_pred)
    r2_test = r2_score(y_test, y_test_pred)
    results.append([name, mse_train, r2_train, mse_test, r2_test])
models_df = pd.DataFrame(results, columns=['Model', 'Train MSE', 'Train R2', 'Test MSE', 'Test R2'])
# === GUI Section (INTERFACE) ===
def predict_and_plot():
    try:
        pH = float(entry_ph.get())
        temp = float(entry_temp.get())
        time = float(entry_time.get())
    except ValueError:
        messagebox.showerror("Input Error", "Please enter valid numeric values.")
        return
    user_data = pd.DataFrame([[pH, temp, time]])
    predictions = []
    for model in models.values():
        predictions.append(model.predict(user_data)[0])
    # Show prediction results
    for i, val in enumerate(predictions):
        prediction_labels[i]['text'] = f"{val:.2f} %"
    # Plot predictions
    fig, ax = plt.subplots(figsize=(6, 4))
    ax.bar(model_names, predictions, color=['skyblue', 'orange', 'green', 'purple'])
    ax.set_title("Predicted Protein Content by Model")
    ax.set_ylabel("Protein Content (%)")
    ax.set_ylim(0, max(predictions) * 1.2)
    ax.grid(axis='y', linestyle='--', alpha=0.7)
    for widget in frame_plot.winfo_children():
        widget.destroy()
    canvas = FigureCanvasTkAgg(fig, master=frame_plot)
    canvas.draw()
    canvas.get_tk_widget().pack()
```

```
# Tkinter window
root = tk.Tk()
root.title("Protein Content Prediction - ML Models")
root.geometry("900x650")
# === R² and MSE (Mean Square Error) of all the models are displayed in the form of Table (Top section) ===
frame_top = tk.Frame(root)
frame_top.pack(pady=10)
tk.Label(frame_top, text="Model Evaluation Metrics", font=('Helvetica', 16, 'bold')).pack()
tree = ttk.Treeview(frame_top, columns=list(models_df.columns), show='headings', height=5)
for col in models_df.columns:
   tree.heading(col, text=col)
   tree.column(col, anchor=tk.CENTER)
for _, row in models_df.iterrows():
   tree.insert('', tk.END, values=list(row))
tree.pack()
# === Middle Section: Input + Predictions ===
frame_mid = tk.Frame(root)
frame_mid.pack(pady=20)
tk.Label(frame_mid, text="Enter Input Features:", font=('Helvetica', 14)).grid(row=0, column=0, columnspan=2)
tk.Label(frame_mid, text="pH:").grid(row=1, column=0, sticky="e")
entry_ph = tk.Entry(frame_mid)
entry_ph.grid(row=1, column=1)
tk.Label(frame_mid, text="Temperature (°C):").grid(row=2, column=0, sticky="e")
entry_temp = tk.Entry(frame_mid)
entry_temp.grid(row=2, column=1)
tk.Label(frame_mid, text="Time (min):").grid(row=3, column=0, sticky="e")
entry_time = tk.Entry(frame_mid)
entry_time.grid(row=3, column=1)
```

```
tk.Button(frame_mid, text="Predict", command=predict_and_plot, bg="green", fg="white").grid(row=4, column=0, columnspan=2, pady=10)
```

# === Display Predictions ===

```
model_names = list(models.keys())

prediction_labels = []

for i, model_name in enumerate(model_names):

    tk.Label(frame_mid, text=f"{model_name} Prediction:").grid(row=5+i, column=0, sticky="e")

    label = tk.Label(frame_mid, text="-")

    label.grid(row=5+i, column=1, sticky="w")

    prediction_labels.append(label)
```

# === Bottom Section: Plot ===

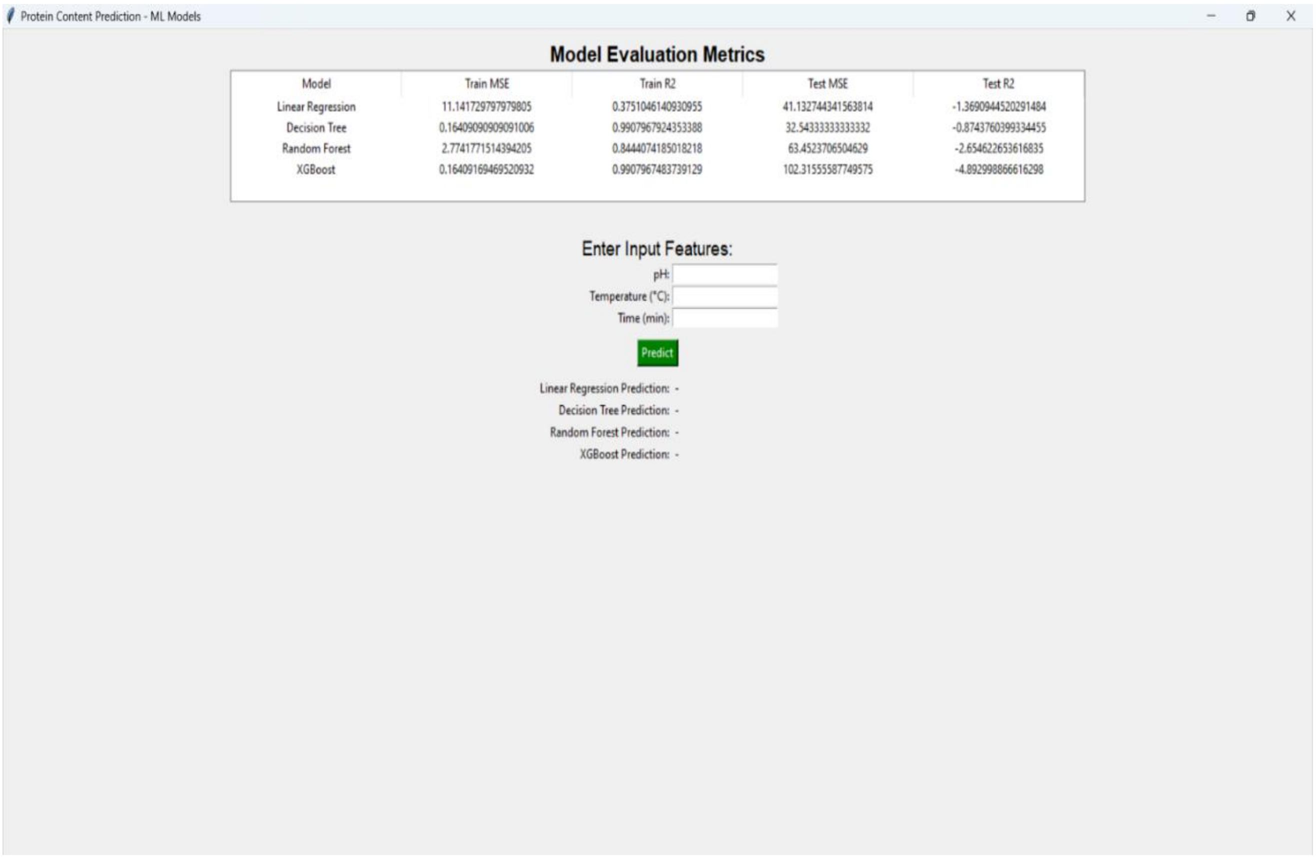
```
frame_plot = tk.Frame(root)

frame_plot.pack(pady=10)

root.mainloop()
```

OUTPUT

The interface that we will obtain is as follows:



Protein Content Prediction - ML Models

Model Evaluation Metrics

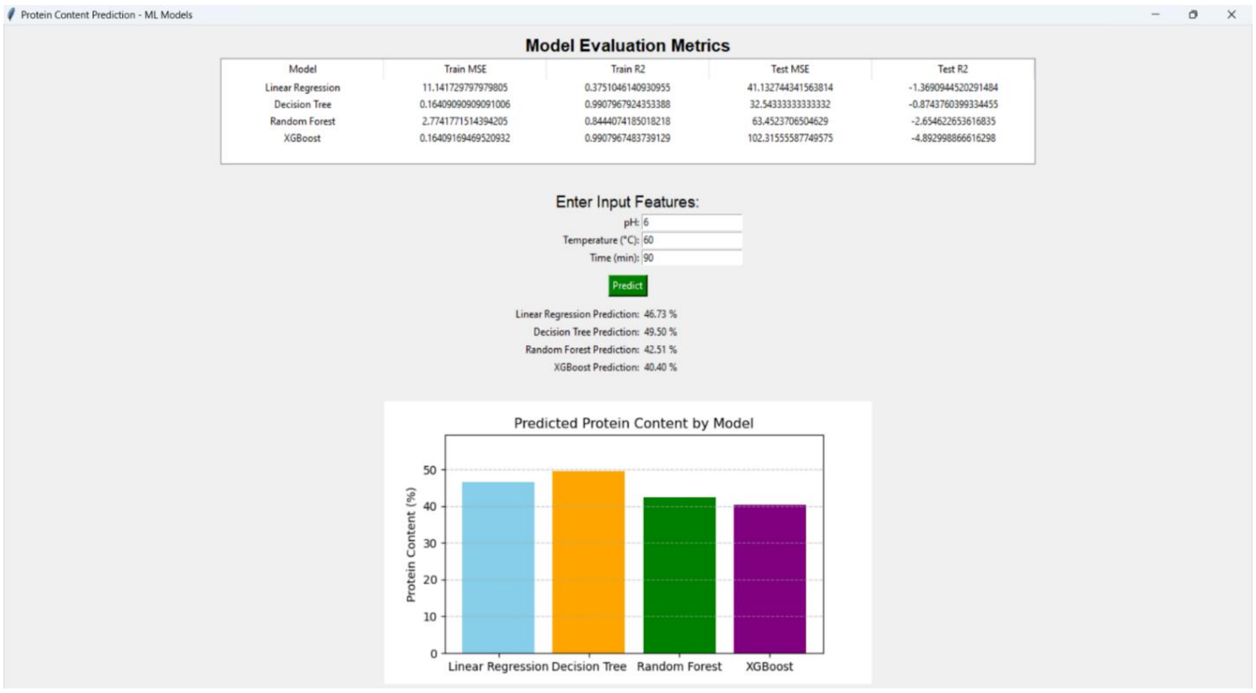
Model	Train MSE	Train R2	Test MSE	Test R2
Linear Regression	11.141729797979805	0.3751046140930955	41.132744341563814	-1.3690944520291484
Decision Tree	0.16409090909091006	0.9907967924353388	32.54333333333332	-0.8743760399334455
Random Forest	2.7741771514394205	0.8444074185018218	63.4523706504629	-2.654622653616835
XGBoost	0.16409169469520932	0.9907967483739129	102.31555587749575	-4.892998866616298

Enter Input Features:

pH:
Temperature (°C):
Time (min):

Predict

Linear Regression Prediction: -
Decision Tree Prediction: -
Random Forest Prediction: -
XGBoost Prediction: -



Model Evaluation Metrics

Model	Train MSE	Train R2	Test MSE	Test R2
Linear Regression	11.141729797979805	0.3751046140930955	41.132744341563814	-1.3690944520291484
Decision Tree	0.16409090909091006	0.9907967924353388	32.54333333333332	-0.8743760399334455
Random Forest	2.7741771514394205	0.8444074185018218	63.4523706504629	-2.654622653616835
XGBoost	0.16409169469520932	0.9907967483739129	102.31555587749575	-4.892998866616298

Enter Input Features:

pH: 6
Temperature (°C): 60
Time (min): 90

Linear Regression Prediction: 46.73 %
Decision Tree Prediction: 49.50 %
Random Forest Prediction: 42.51 %
XGBoost Prediction: 40.40 %

The interface consists of three sections:

1. **Top section**

The top section consists of R^2 and MSE (Mean Square Error) of all the models are displayed in the tabular form to compare which model is the best fit.

2. **Middle Section**

It consists of the input section as well as the prediction.

3. **Bottom Section**

The graph plotted between the predicted values of all the models is displayed.

Chapter-4

RESULT AND DISCUSSION

The research study involved the comparison of various regression models, i.e., Linear Regression, Decision Tree Regressor, Random Forest Regressor, and XGBoost Regressor, on a data set with the defining predictors of pH, temperature, and processing time, and the target variable as protein content (%). The performance metrics derived from this analysis easily revealed that each model exhibited different characteristics. Linear Regression was the control model, providing the first insight into the underlying linear relationships in the data. But it was seen that advanced models like Random Forest and XGBoost outperformed the simple linear model with their lower Mean Squared Error (MSE) and higher R^2 values, thus better capturing the non-linear trends and complex interactions between the input variables. Tabular display of performance metrics in the GUI was an essential aspect for comparative analysis. It facilitated instantaneous identification of strengths and weaknesses of single models. The Decision Tree model, being interpretable, was more inconsistent due to its overfitting tendency, particularly evident in training vs. testing measure comparisons. Ensemble approaches such as Random Forest and XGBoost, however, circumvented this problem through their inherent aggregation process, producing more stable predictions across data partitions. These results are in line with the traditional performance of regression model literature, which indicates the robustness of ensemble approaches in dealing with complicated real-world datasets. In addition to the quantitative findings, the graphical user interface (GUI) of the project was also a significant contributor to demonstrating the appropriateness of the models. By way of real-time input of significant features and concurrent visual display of predicted protein content as bar charts, the GUI was an effective interface between algorithmic model performance and user-friendly application. Users are able to visually identify the relative performance distinctions of the models and thus increase the reliability of the ensemble methods in making vital predictions in quality control and biotechnological process development. The argument also favors the selected methodology by discussing possible weaknesses and suggesting avenues for future enhancement. While existing results verify the superiority of ensemble approaches in this context, there are prospects for future refinement in the form of hyperparameter tuning and more feature engineering, which will improve predictive accuracy. Overall, the synergy between quantitative and qualitative analyses offers compelling evidence for the project methodology, proving that the application of sophisticated regression models in an interactive setting can effectively facilitate decision-making processes in protein content prediction and other pertinent applications.

Chapter-5

FUTURE PERSPECTIVE AND CONCLUSION

Innovation in food plant-based The Future of Artificial Intelligence Concerning Plant-Based Alternatives for Protein Extraction:

Better Texture and Taste: AI algorithms are capable of predicting the behavior of extracted proteins in various conditions and thus enable a close imitation of animal proteins like meat and dairy in texture which immensely matters for consumer penetration into such plant-based products.

Nutrition Personalization: Using AI, nutritional completeness in products can be achieved by blending various oilseed proteins (for example, soy, sunflower, canola) via analyzing amino acid profiles. This guarantees health benefits from plant food equivalent and to complement animal protein in protein requirements.

Cost efficiency and optimized resources: While making plant-based protein extraction far more economically feasible, AI can improve efficiency on a grand scale:

Reduced Input Usage: It would determine better process parameters such as solvent concentration, temperature, and time of extraction thereby requiring less water and energy as well as chemicals while maintaining the yield.

Byproduct Utilization: Residues from the extraction like fibers and hulls can be delivered by AI to come up with possible uses as ingredients in bakery products, snacks, or supplements to mitigate waste and increase revenues for a company.

Functional Protein Development: AI therefore has the capability of customizing proteins for specific functionality within a food application:

Target Functionality: Predictive modeling correlates the protein structure with its functional behavior such as solubility and emulsifying and gel-forming properties. This makes it possible to engineer proteins for certain functionalities, such as:

- Emulsifiers suited for creamy dressings

- Gelling agents formulated for plant-based meat textures
- Foaming agents for use in whipped toppings or baked goods

Product-Specific Tailoring: AI develops ingredient combinations that are optimum in a specified food matrix, be it a high-protein snack bar or a dairy-free yogurt.

Sustainability and Circular Economy: AI enables sustainable operations, smarter processing methods:

Energy-efficient extraction pathways are predicted thanks to machine learning models so that no energy consumption is wasted, bringing down carbon footprint.

Water-holding and efficient AI-based systems can be deployed to track the uses of water in real-time, identify excess uses, and suggest places to recycle water from.

Innovation in byproducts: AI helps in identifying those compounds in waste streams that can be turned into bioactive ingredients, supplements, or even biodegradable packaging materials, thus taking the goal to the next level in the circular economy.

Real-time Quality Assurance: Data from smart sensors: As well as monitoring and recording all relevant key variables like temperature, pH, and viscosity, AI interventions would use this data to effect real-time adjustment in all operations in their most optimal venue and thus eliminate all wastage, errors, and inefficiencies.

Batch Consistency: The AI algorithms work to minimize the variability in production, which ultimately means that every batch has to fall within the predetermined parameters concerning protein content purity and safety.

Early Detection of Defects: AI-influenced systems could help prevent food loss and safeguard food safety by detecting an anomaly such as bacterial contamination or mechanical failures.

Accessibility and Global Scale: The science democratization of AI delivers region-wide accessibility and scales nicely:

Small Scale Configurations: Mobilization or modularization of AI systems will configure then for use with rural or small-scale food processors as this enhances access to high-quality plant proteins.

Regional Adaptation: AI could optimize the processes upon learning from the local environmental parameters like humidity and temperature then also from the regional varieties of crops like mustard seeds in India as well as rapeseeds in Europe.

Global Food Security: On the larger scale, while not entirely eliminating dependency in animal agriculture, AI solutions are used in providing stable and affordable sources of protein beefing up global nutrition.

Research and Experimental Advancement

The novel machine provides paths for improvements in research and innovation in food technology:

Experimental Design: AI will simulate several extraction conditions to help the researcher center attention on the most promising on-bench trials.

Data-Driven Insights: Hidden patterns, correlations, and trends derive from analyzing large datasets crossing various experiments to realize new insights into extraction techniques, novel food products, and ingredient combinations.

This means that such development cycles will be speeded up, with opportunities transforming quickly from idea to market, allowing food companies to be nimbler in responding to consumer trends and dietary needs (keto, vegan, high protein, etc.).

CONCLUSION

This project is a practical and structured realization with several machine-learning models that predict the protein content, with a GUI developed in Python using Tkinter. Its main object was to develop a predictive system with the ability to estimate the protein content based on values of user input like pH, temperature, and time. The project illustrates the efficacy of machine learning in practical situations, mainly in fields like food analysis and quality control.

For the system to work, a dataset consisting of various features affecting protein content was initially used. The preprocessing of data included feature and label separation, conversion into relevant data types, and train-test split to ensure effective generalization by the models. The data were trained via four different regression algorithms: Linear Regression, Decision Tree Regressor, Random Forest Regressor, and XGBoost Regressor. Each was trained on this data because the particular models have their own strengths concerning specific data structures.

The model performance is evaluated based on core evaluation parameters such as Mean Squared Error (MSE) and r-square (R^2) format pertaining to train and test set performance. This analysis gives a sound insight into how well these models perform in predicting protein content and the level to which they generalize in unforeseen situations. A tabulated comparison summation of this measure was set at the upper end of the interface for the users to analyze the performance difference among models with ease.

Most importantly, the interface is extremely user-friendly and does not require programming skills to input the experimental conditions: namely pH, temperature, and time; this interface has been very intuitively designed with Tkinter. The upper part consists of a table showing evaluation metrics; the center part collects user input along with prediction results; the bottom section shows the result in bar charts, which compare predictions from each model. When a user inputs the required parameters and presses the prediction button, the system will examine whether the format of the input is correct, and protein content prediction will be done using all four models. The result along with the names of each model is displayed clearly, and a color-coded bar chart shows a visual comparison helping users to see which model gives high or low prediction for the given input conditions.

This is a huge assistant in practical application domains especially in food quality control or research. It helps save time for lab technicians, students, or food scientists-to determine protein content using process parameters, as it reduces the number of experiments typically carried out on these outputs. Most importantly, this project lays a good groundwork for all sorts of development concerning future improvements such as additional input variable.

Overall, the project demonstrates the smooth joining of machine-learning with ease-of-use software design. It emphasizes converting highly complex analytical tools into user-friendly applications that aid data-driven decision-making for scientific and industrial purposes.

REFERENCES

- Bishop, C. M. (2006). Pattern recognition and machine learning. New York, NY: Springer.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY: ACM. https://doi.org/10.1145/2939672.2939785
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: With applications in R. New York, NY: Springer.
- Grayson, M. (2000). Python and Tkinter programming. Birmingham, UK: O'Reilly Media.
- Quinlan, J. R. (1986). **Induction of decision trees**. *Machine Learning*, 1, 81–106. https://doi.org/10.1023/A:1022643204877
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). **Introduction to linear regression analysis** (5th ed.). Wiley.
- Kuhn, M., & Johnson, K. (2013). **Applied predictive modeling**. Springer. https://doi.org/10.1007/978-1-4614-6849-3
- Chen, T., & Guestrin, C. (2016). **XGBoost: A scalable tree boosting system**. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785
- Gholami, A., & Esfahani, S. S. (2021). Use of artificial intelligence in the circular economy of food production: A case study on plant-based protein. Journal of Cleaner Production, 297, 126602. https://doi.org/10.1016/j.jclepro.2021.126602
- Brownlee, J. (2019). Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End. Machine Learning Mastery.
- Tkinter Documentation – Python Software Foundation. (n.d.). Retrieved from https://docs.python.org/3/library/tkinter.html
- https://pmc.ncbi.nlm.nih.gov/articles/PMC11095727/?utm_source=chatgpt.com
- https://www.sciencedirect.com/science/article/abs/pii/S0889157523008396?utm_source=chatgpt.comhttps://pubmed.ncbi.nlm.nih.gov/39775201/
- https://onlinelibrary.wiley.com/doi/full/10.1002/pro.4467?utm_source=chatgpt.com
- https://pubmed.ncbi.nlm.nih.gov/22101336/