Introduction

Personalized diets are becoming increasingly important as nutrition science evolves beyond one-diet-fits-all recommendations. Factors such as age, body composition, dietary preferences, and health conditions all influence how individuals respond to different foods. With rising awareness of chronic diseases like obesity, diabetes, and cardiovascular issues, the need for tailored dietary plans has been ever more urgent. Personalized nutrition aims to optimize health outcomes by aligning food choices with individual characteristics and lifestyles.

This project explores the use of machine learning to predict recommended meal plans based on user-specific data. The dataset I use is from Kaggle and features patient health profiles, with data on demographic and physiological features such as age, BMI, fat and protein intake, and dietary habits. My goal is to train predictive models that can classify individuals into one of several meal plan categories: Balanced, High-Protein, Low-Carb, or Low-Fat diets. Multiple algorithms — including logistic regression, decision trees, K-nearest neighbors, and neural networks — are evaluated to determine which best captures the complex, potentially nonlinear relationships between personal traits and dietary recommendations.

This project is meaningful because it demonstrates how data-driven methods can support more informed, individualized health decisions. By identifying which models perform best, I'm excited to explore the growing field of personalized nutrition and learn about adaptive dietary planning tools in the future.

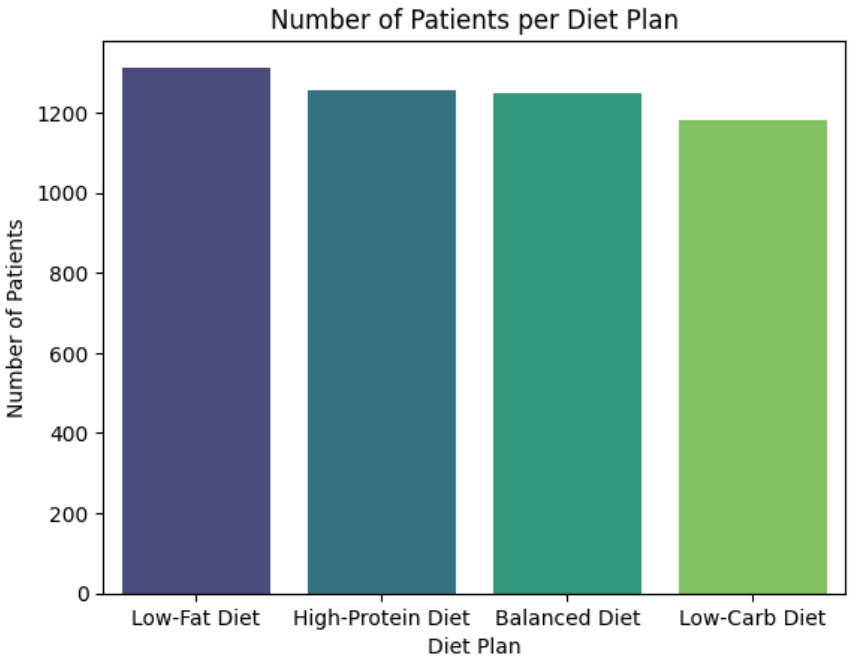
Preprocessing the Dataset

Upon first glance, this dataset needed some preprocessing before we jump into EDA and predictive modelling. It contained 5000 entries in total and 30 columns. I identified columns with null values – "Allergies", "Chronic_Disease", and "Food_Aversions", and proceeded to fill the null values. I also dropped irrelevant data that won't be useful for deriving relationships between recommended meal plan and patient health profile, like Patient ID and "Recommended Carbs/Fats/Calories/Proteins".

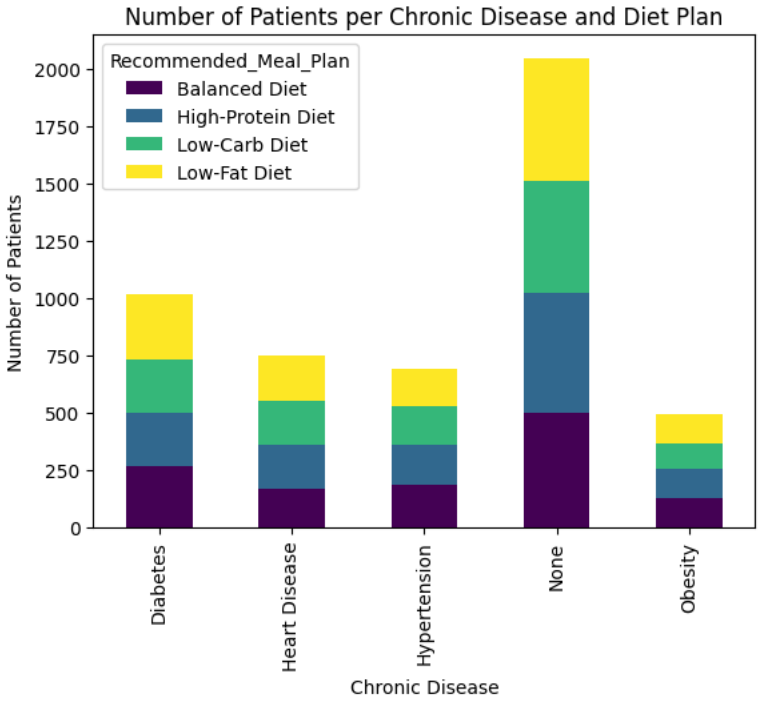
The unique values for a recommended diet plan are: High-Protein Diet, Balanced Diet, Low-Fat Diet, and Low-Carb Diet. I will be using both qualitative and quantitative data spanning medical history, lifestyle, dietary intake, and food preferences and restrictions to classify patients into a personalized diet plan. For the predictive modelling section, I also dropped duplicates, outliers, and highly correlated columns in the dataset to improve prediction accuracy

Exploratory Data Analysis

For exploratory data analysis, I first skimmed the distribution of patients under each diet plan category.



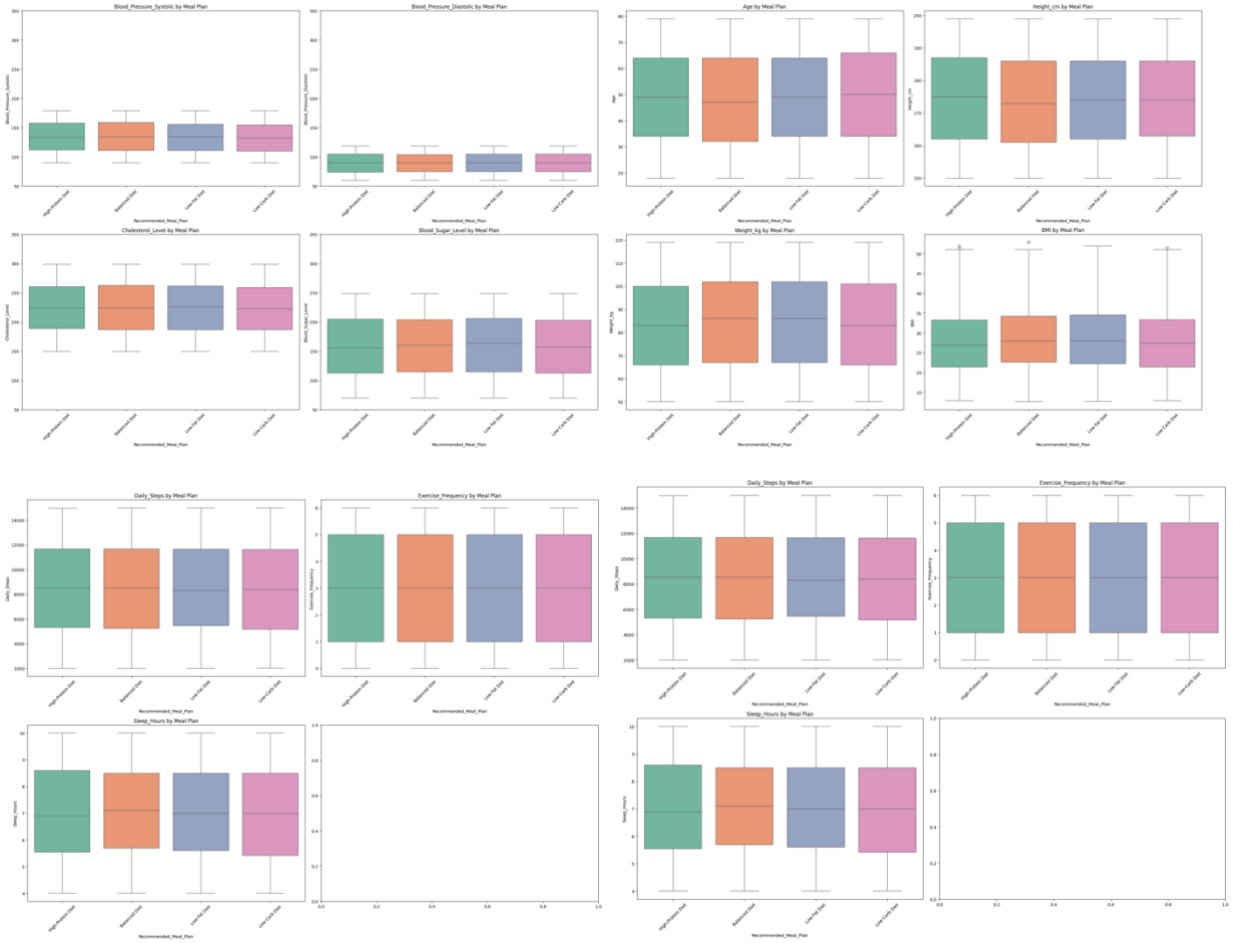
Turns out, the distribution is fairly even, with most patients falling under the Low-Fat Diet category.



Wondering if there would be a pattern between chronic diseases and the diet plans these patients are recommended, I also plotted the distribution of patients under each chronic disease category and diet plan. Every bar seems to have very even distribution of meal

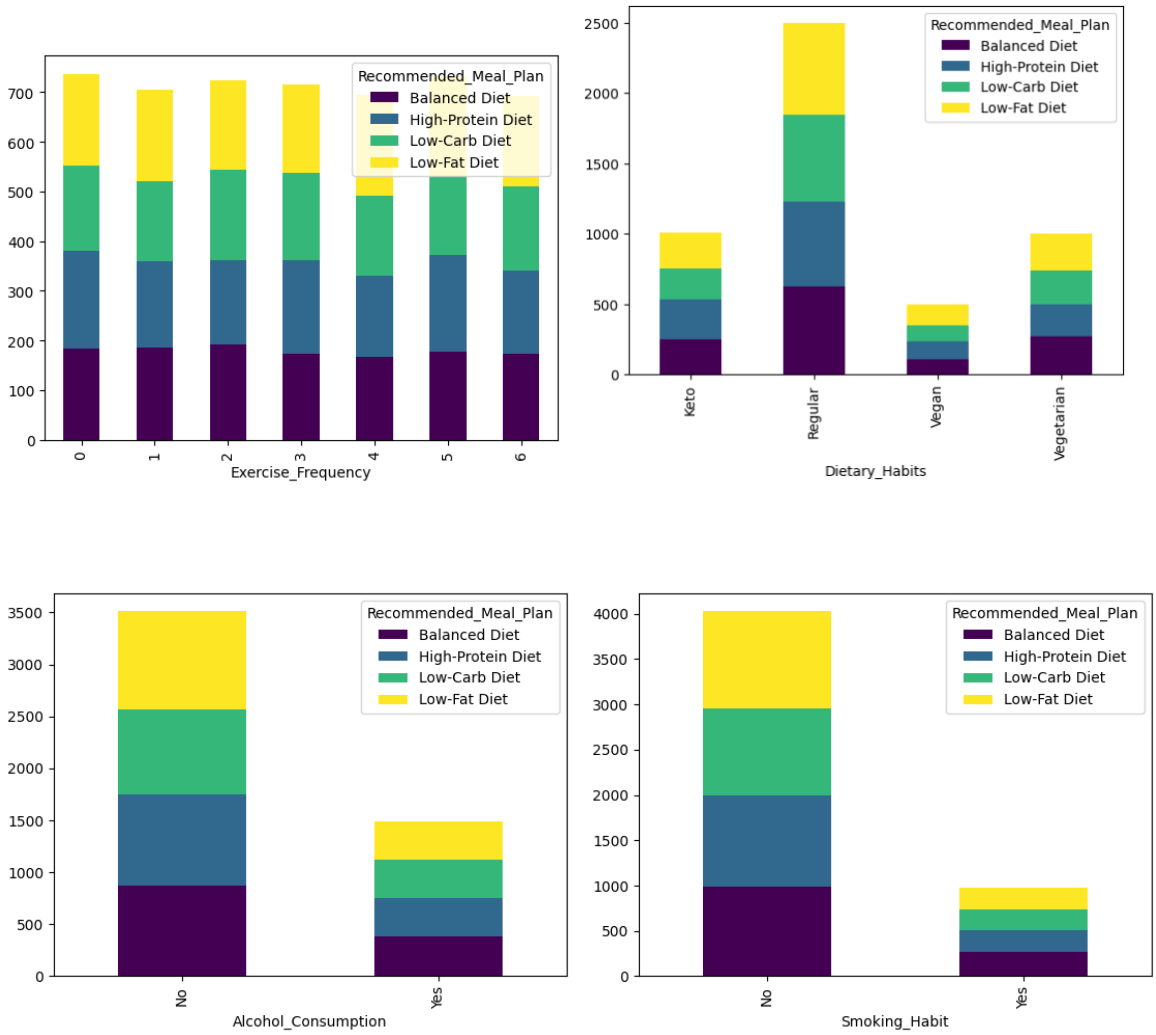
plans, suggesting that the type of chronic disease alone isn't a strong, direct classifier of diet plan.

Since this dataset is separated into 4 main parts: Medical history, Demographics, Lifestyle, and Dietary habits, I plotted the distribution of numerical stats within each category against diet plan.



As the side by side plots illustrate, there aren't noticeably large differences in the distribution of these numerical features against diet plan at all; the medians and IQRs are quite close. Compared to the rest, the category of Demographics data showed the most variation, but that's probably because the collection of patients are very diverse.

Further, I plotted categorical features of Lifestyle habits (including Smoking Habit, Alcohol Consumption, Exercise Frequency, Dietary Habits) against Recommended Meal Plan.



Again, the distribution of recommended diet plans are even across all bars. This tells me that the relationship between both numerical and categorical features and the recommended meal plan has a potentially non-linear relationship, and that even though no singular feature is a strong predictor, perhaps it's the combination of many factors that classifies a patient's meal plan. Taking this insight with me, I moved onto predictive modelling, hoping that perhaps, nonlinear models like decision trees can detect subtler patterns.

Predictive Models and Methods

My goal is to predict the categorical feature "Recommended Meal Plan", so I will be using accuracy score and classification reports to measure my models' performance. As this is a classification as opposed to regression problem, the models I tested are: Multiple Regression, KNN, Decision Tree, and Neural Networks.

For all my models, I made sure to separate the dataset into numerical and categorical features and transformed my categorical features with OneHotEncoder and scaled numerical features with StandardScaler. One concern was that since there are so many predictors, the models might become overcrowded and confused by the noise; therefore, I decided to run regression, KNN, and decision tree on all the features vs most significant features to see if accuracies would improve.

1. Creating a Baseline Model

For my baseline model, I chose to use a DummyClassifier with a random state = 42 and uniform strategy; I specifically chose a uniform strategy because it tends to be useful when there is no specific pattern to guide the predictions, and from our EDA, there is no strong pattern or predictor.

The baseline model's accuracy score is around 23%, which makes sense since it is close to the probability of randomly predicting ¼ diet plans (25%). Low-Fat Diet has the highest precision score of 26%. Since it is more important to minimize the health risks of recommending the wrong type of diet, I would be mostly analyzing precision scores.

	precision	recall	f1-score	support
Balanced Diet	0.21	0.22	0.21	251
High-Protein Diet	0.24	0.22	0.23	252
Low-Carb Diet	0.19	0.21	0.20	216
Low-Fat Diet	0.26	0.26	0.26	281
accuracy			0.23	1000
macro avg	0.23	0.23	0.23	1000
weighted avg	0.23	0.23	0.23	1000

2. Multiple Regression Model

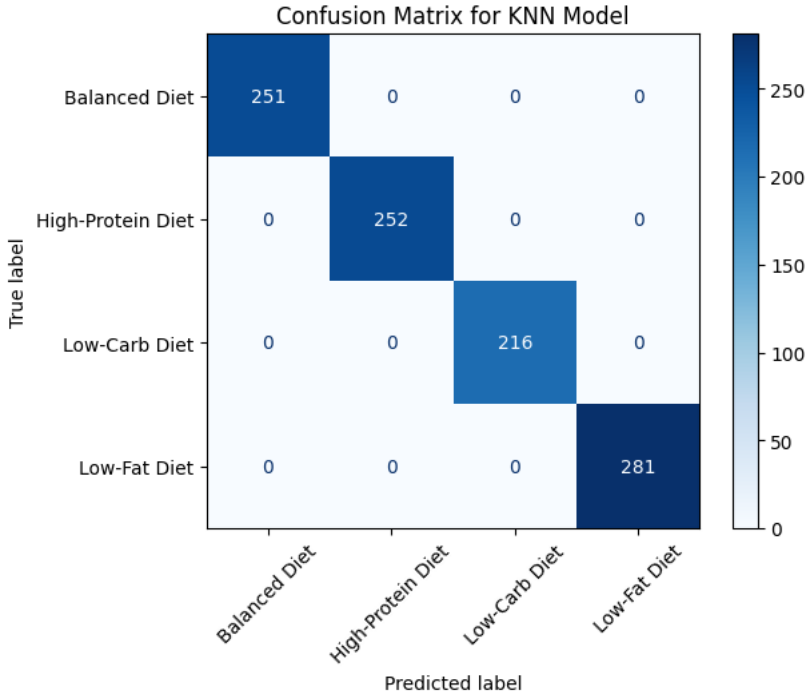
I started with a Multiple Regression model with a 80/20 train/test split to explore whether there would be combined effects from independent variables on the recommended meal plan. Unfortunately, this model returned a test accuracy score of 23% - equal to baseline prediction accuracy – yet an almost 30% accuracy score on the train data. This gap between train and test accuracy, especially when test accuracy is lower, suggests that my model may be overfitted.

	precision	recall	f1-score	support
Balanced Diet	0.26	0.28	0.27	251
High-Protein Diet	0.28	0.26	0.27	252
Low-Carb Diet	0.22	0.17	0.19	216
Low-Fat Diet	0.28	0.32	0.30	281
accuracy			0.27	1000
macro avg	0.26	0.26	0.26	1000
weighted avg	0.26	0.27	0.26	1000

The Multiple Regression Model shows higher than Baseline precision scores, but they are still alarmingly low. The precision scores for Balanced and Diet and High-Protein Diet increased considerably from the baseline performance, and Low-Fat Diet remains the highest performing meal plan category.

3. K-Nearest Neighbors Model

Next, I used a KNN model because it is a simple, adaptable classifier and could handle non-linear relationships in data. The best parameter KNN model with Grid Search performed marginally better than the Multiple Regression Model, with a test accuracy score of 0.281 and train accuracy score of 1.0. This confirms that there is a nonlinear relationship between predictors and the target, since KNN doesn't assume any specific relationship and can capture complex, non-linear relationships.



However, overfitting is definitely a problem – there is likely too many random fluctuations and noise in the data, and that there's a high variance in predictions. KNN is more appropriate for a smaller number of inputs, so I will refit the KNN model with fewer number of features later.

4. Decision Tree Classifier

I then move onto the Decision Tree model, which is also effective for capturing non-linear relationships. This time, the test accuracy score was nearly 56%, a significant improvement from my baseline model and all the other models I've tried so far. It is also close to the train accuracy score of 55.6%, suggesting that with the Decision Tree classifier, the overfitting problem might be resolved. This could potentially be explained by the fact that Decision trees work better with large datasets, and that they tend to identify and incorporate feature interactions, an aspect that KNN requires explicit feature engineering for.

	precision	recall	f1-score	support
Balanced Diet	0.70	0.47	0.57	251
High-Protein Diet	0.50	0.57	0.53	252
Low-Carb Diet	0.54	0.56	0.55	216
Low-Fat Diet	0.54	0.61	0.57	281
accuracy			0.56	1000
macro avg	0.57	0.55	0.55	1000
weighted avg	0.57	0.56	0.56	1000

Interesting observation: Precision score for Balanced Diet is the highest instead of Low-Fat Diet

Retesting Multiple Regression, KNN, and Decision tree with a narrowed down feature list to reduce overfitting

Since there were consistently large gaps between my train and test accuracies, I decided to use fewer features to predict my target. From the lists of most important features from the previous models, I chose overlaps in 4-5 features. This will hopefully help reduce dimensionality, which improves performance and interpretability, speed up training and testing with a large dataset, and reduces overfitting risk, especially in the KNN model.

LGR Model

Permutation Importance:

	Feature	Importance Mean	Importance Std
15	Dietary_Habits	0.012946	0.007974
3	Chronic_Disease	0.006740	0.007937
16	Caloric_Intake	0.004404	0.004391
19	Fat_Intake	0.004338	0.005047
11	Exercise_Frequency	0.003770	0.004076
13	Alcohol_Consumption	0.002736	0.004903
2	BMI	0.002736	0.009005
21	Food_Aversions	0.002636	0.006500
5	Blood_Pressure_Diastolic	0.001902	0.003432
8	Genetic_Risk_Factor	0.001835	0.003852
20	Preferred_Cuisine	0.001768	0.005743
1	Gender	0.001735	0.006283
17	Protein_Intake	0.001235	0.005993
12	Sleep_Hours	0.001001	0.005322
0	Age	0.000567	0.005875
10	Daily_Steps	0.000534	0.003579
18	Carbohydrate_Intake	-0.001301	0.003583
14	Smoking_Habit	-0.001602	0.005388
9	Allergies	-0.002369	0.005868
6	Cholesterol_Level	-0.002469	0.004888
7	Blood_Sugar_Level	-0.002669	0.003801
4	Blood_Pressure_Systolic	-0.003036	0.005534

Decision Tree

	0
BMI	0.013814
Cholesterol_Level	0.010210
Protein_Intake	0.005806
Age	0.002002
Gender	0.000000
Chronic_Disease	0.000000
Blood_Pressure_Diastolic	0.000000
Blood_Pressure_Systolic	0.000000
Genetic_Risk_Factor	0.000000
Blood_Sugar_Level	0.000000
Daily_Steps	0.000000
Exercise_Frequency	0.000000
Sleep_Hours	0.000000
Allergies	0.000000

KNN

	0
Sleep_Hours	0.028028
Age	0.022322
Blood_Sugar_Level	0.022222
Carbohydrate_Intake	0.019520
Preferred_Cuisine	0.019319
Chronic_Disease	0.019119
BMI	0.018819
Blood_Pressure_Diastolic	0.018018
Fat_Intake	0.017718
Food_Aversions	0.016016
Cholesterol_Level	0.013313
Exercise_Frequency	0.013313
Caloric_Intake	0.013013
Gender	0.012412
Blood_Pressure_Systolic	0.011812
Alcohol_Consumption	0.011111
Genetic_Risk_Factor	0.007908
Daily_Steps	0.007808
Dietary_Habits	0.007608
Protein_Intake	0.002903

Looking at these lists, 5 features seem to consistently perform decently across the three models:

1. Chronic disease
2. Fat intake
3. BMI
4. Food Aversions
5. Blood Pressure Diastolic

```
=== Model Accuracy Comparison ===
Logistic Regression: 0.265
KNN: 0.283
Decision Tree: 0.255
```

Unlike my predictions, the accuracy scores for each model did not go up.

5. Neural Networks Model

Since narrowing down the features list didn't seem to improve the accuracy of my best fitted model (decision tree model), I decided to take the top 10 important features from my previous decision tree model and fit them on a neural network model. Neural networks thrive on multiple features, so I'm hoping this would be the best performing model out of all.

Feature List used: Protein_Intake, Caloric_Intake, BMI, Daily_Steps, Carbohydrate_Intake, Fat_Intake, Cholesterol_Level, Age, Blood_Sugar_Level, Blood_Pressure_Diastolic

I trained the model with 3 layers vs 4 layers to see if accuracies would improve from a more complex model:

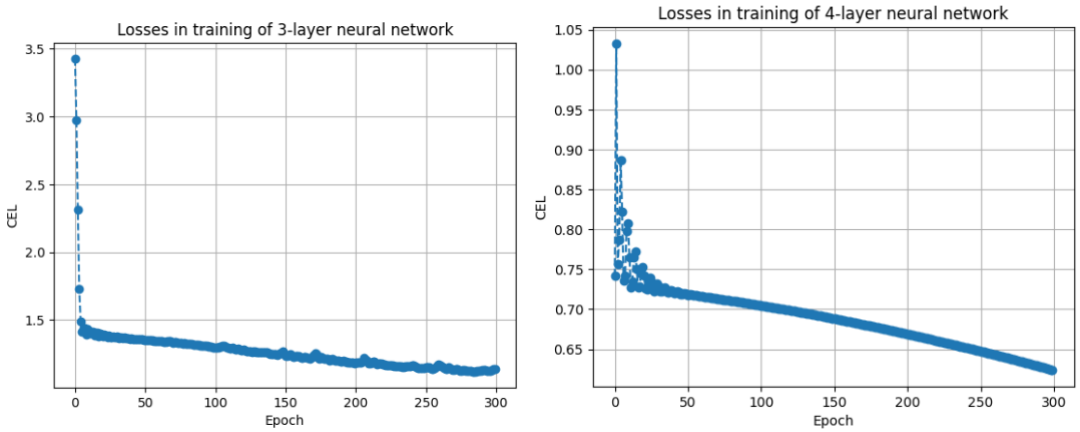
1. Classification Report With 3 layers, lr = 0.1 (Left) vs 4 layers, lr=0.001 (Right)

	precision	recall	f1-score	support
0	0.42	0.67	0.52	1248
1	0.56	0.41	0.47	1254
2	0.52	0.49	0.51	1182
3	0.60	0.46	0.52	1312
accuracy			0.50	4996
macro avg	0.53	0.51	0.50	4996
weighted avg	0.53	0.50	0.50	4996

	precision	recall	f1-score	support
0	0.85	0.89	0.87	1248
1	0.64	0.64	0.64	1254
2	0.63	0.56	0.59	1182
3	0.83	0.87	0.85	1312
accuracy			0.74	4996
macro avg	0.74	0.74	0.74	4996
weighted avg	0.74	0.74	0.74	4996

Precision scores significantly improved

2. Cross Entropy Loss with 3 layers vs 4 layers



Comparing my losses in training the data for multiclass neural network, the one for 4 layers declined more smoothly than my 3 layer neural network model -the 3 layer neural network model loss also showed decline in loss overtime, but showed small spikes in the tail, signaling an unstable model. After adjusting the learning rates, the 4 layer model exhibited a smooth decline.

Summary Performance of Neural Network model:

1. Accuracy: 74% - This is well above random guessing for 4 classes (random chance = 25%), indicating that my model is learning meaningful patterns.

2. Macro avg F1-score: 0.74 – This treats all classes equally, so a good macro score means my model is not heavily favoring one class

3. **Class 0 and 3 perform best** – Precision and recall are both **above 0.83**, meaning the model is confidently and correctly identifying these classes. **Class 2 needs improvement**: Recall is 0.56, meaning it misses quite a few actual class 2s.

Conclusion

This project explored the potential of using machine learning to predict personalized diet recommendations based on individual health and lifestyle features. Through extensive data exploration and model experimentation—including logistic regression, K-nearest neighbors, decision trees, and a multi-class neural network—we found that non-linear models, particularly neural networks and decision trees, were more effective at capturing the complex relationships between features and diet type. Our best-performing model achieved a significant accuracy improvement over the baseline, indicating that personalized dietary predictions are feasible with the right model and features.

While the classification report showed strong performance for some diet classes, others—particularly class 2—had lower precision and recall, suggesting areas for further improvement. Overall, this project demonstrates the value of data-driven dietary recommendations and lays the foundation for more personalized nutrition solutions.

Next Steps

With more time, I would explore:

1. **Improve Class Imbalance:** Explore techniques like class weighting to improve predictions for underperforming diet types.
2. **Feature Engineering**: Develop domain-informed features or interactions to strengthen model signal.
3. **Model Tuning:** Further optimize the neural network architecture and hyperparameters (e.g., learning rate schedules, dropout).
4. **Deployability:** Begin exploring how this model could be integrated into a real-world application, such as a diet recommendation app.
5. **Expand Dataset:** Incorporate more diverse or longitudinal data to improve generalizability and personalization.