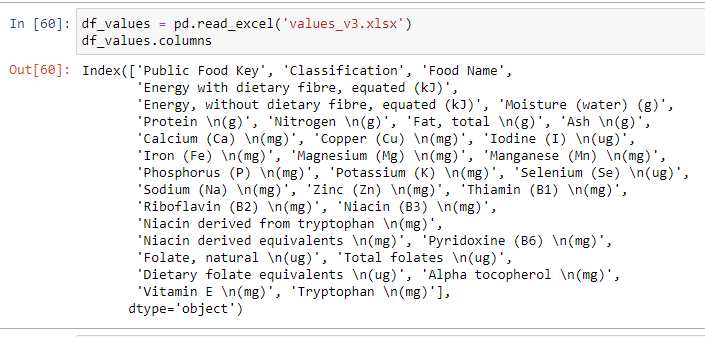
**Data cleaning and data exploration**

Data preparation is critical for machine learning because it can directly affect the final performance of a model. There are 1617 rows and 293 columns in the first table Rel\_2\_Nutrient\_file and second table has 191 rows and 204 columns. I do analysis on the first table Because the second table is a part of the first table. However, most of them are missing value in first table. Some of the features have massive missing values (>1400 missing values each column) like Galactose, Maltotrios, Starch and Dextrin etc. if using mean imputation or random imputation, information of feature will lose, and result may be polluted by mean or random data for they occupy over half of a feature. Therefore, I decided to drop these columns for loss of information integrity. The threshold for deleting features is set to missing values over 400 which means 76% integrity remains in the features can stay in table, which may does not significantly impact to the performance of models.

A screenshot of a computer

Description automatically generated

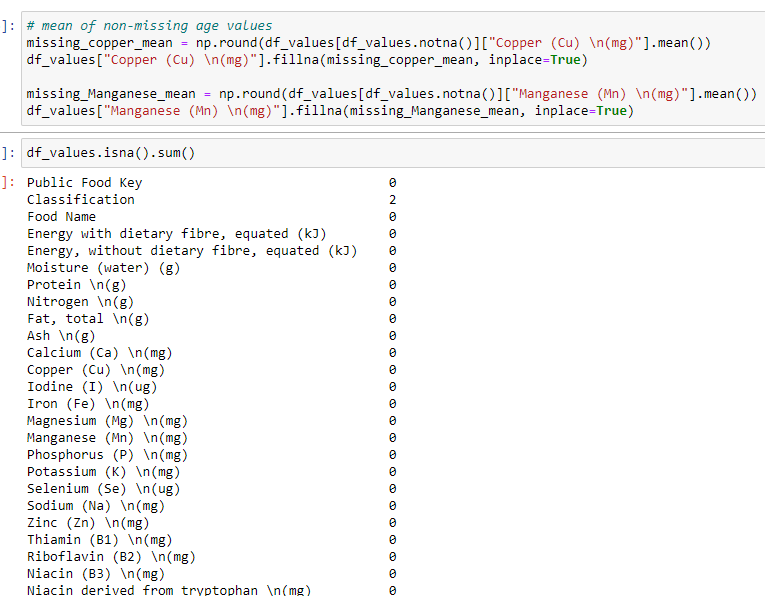
While examining the values inside a column using the pd.describe() function, I noticed that several features contain a significant number of zero values. The presence of numerous zeros within a feature can potentially affect the mean and overall integrity of the information. To solve this issue, I made the decision to drop any column that has more than 400 zero values (same reason of dropping missing values). After implementing this strategy, I obtained the figure below, which displays the remaining columns (33) after the column dropping are executed.



Moving on to the next step, addressing missing values becomes a priority task. To handle this, I opted for Mean imputation when dealing with numerical data. This approach involves replacing the missing values with the mean value of the respective feature. By fill missing numerical data using the mean of each columns, we can preserve the general statistical characteristics of the dataset while ensuring a completer and more reliable dataset for further analysis.

A screenshot of a computer

Description automatically generated with low confidence



Classification data is imputed by random imputation.

A screenshot of a computer program

Description automatically generated with low confidence

Now, data quality preparation is done.

For the remaining features, ‘classification’, ‘public food key’, ‘food name’ are purely categorical features. Other features are numerical can be used to do regression predication.

Using seaborn, I created a correlation matrix for remaining features.

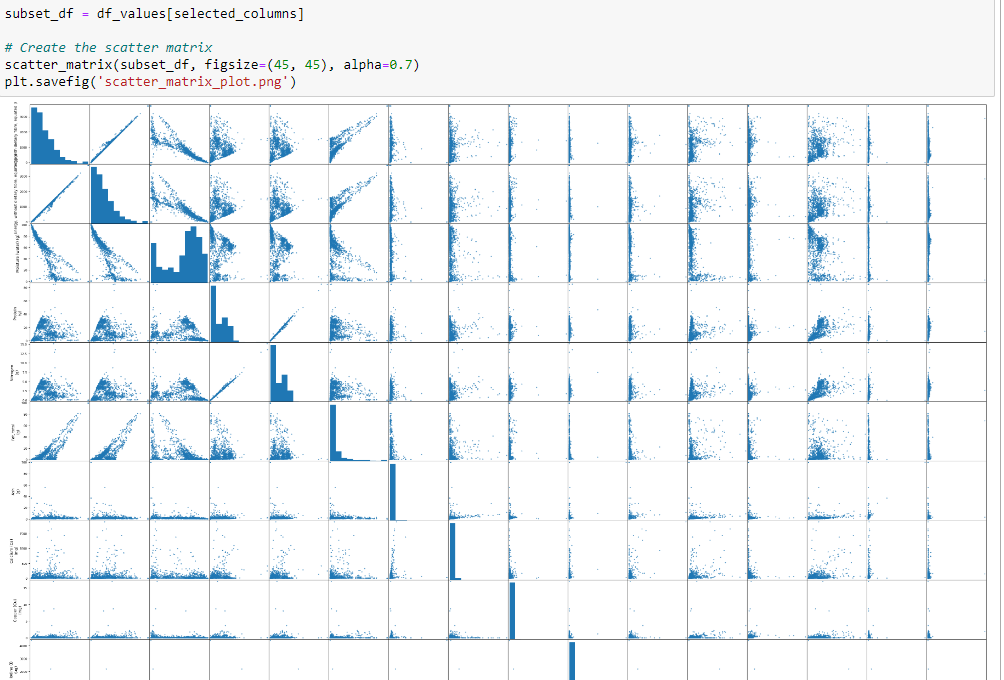
Detailed heat map at appendix

A picture containing text, screenshot, pattern, colorfulness

Description automatically generated

From this heat map, a strong correlation (> 0.8) is observed between ‘fat’ and ‘energy with dietary fibre’. ‘Protein’ and ‘Nitrogen’ are observed to have a perfect positive correlation (1), it indicates they are in perfect synchronization. In the next content, ‘Protein’ and ‘Nitrogen’ have a moderate correlation with phosphorus, it is over 0.6. Meantime, phosphorus has strong relationship (>0.73) with ‘Niacin derived from tryptophan’ and ‘tryptophan’. It is reasonable because Niacin is derived from tryptophan. There is another strong correlation between sodium and ash. The last strong correlation is observed over 0.9 between protein and tryptophan, Nitrogen also has over 0.9 correlation with tryptophan. In negative correlation, moisture has a strong negative correlation (<-0.8) with energy with or without dietary. There are several other moderate correlations of around 0.6-0.7, It is worthwhile to examine the scatter matrix to investigate these correlations.

Detailed scatter plt at appendix



As we observed, most of plots are are clustered around the bottom-left. These are several pairs which have an increase relationship, like ‘alpha tocopherol’ and ‘Vitamin E’, when alpha tocopherol increases, vitamin E also increases. Ash and calcium have steady increase relationship. Nitrogen and protein have sharp increase relationship. Energy and fat have sharp rise relationship.

I aim to create a regression problem in order to apply Machine Learning techniques. For this purpose, I will focus on the following features: 'Energy with dietary fibre, equated (kJ)', 'Moisture (water) (g)', 'Protein (g)', 'Nitrogen (g)', and 'Fat, total (g)'. My objective is to investigate the relationship between fat and the other features to compare the performance of the k-nearest neighbor model (k-nn) and the regression model by evaluating their test accuracy. To enhance the prediction accuracy of each model, I have chosen to utilize Principal Component Analysis (PCA). This method reduces the dimensionality and noisy of remaining features and improves the predication accuracy of each model. Another reason is high correlation of pairs, like ‘Energy with dietary fibre, equated (kJ)’ and ‘Fat’ (>0.8), ‘Moisture (water) (g)’ and ‘Fat’ (>0.5), it affects the performance of regression models, PCA takes advantage of multicollinearity and combines the highly correlated variables into a set of uncorrelated variables. Therefore, PCA can effectively eliminate multicollinearity between features.

Principal Component Analysis (PCA)

I use StandardScaler() in sklearn library to scale input variables to reduce their dimensions while retaining all important information. Z-score is applied in the processing, which is difference of observed and mean of sample divided by standard deviation of sample. Following scree plot and cumulative explained variance plot

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, line, diagram, plot

Description automatically generated

As we see first three principal components capture the almost of information.

A screenshot of a computer

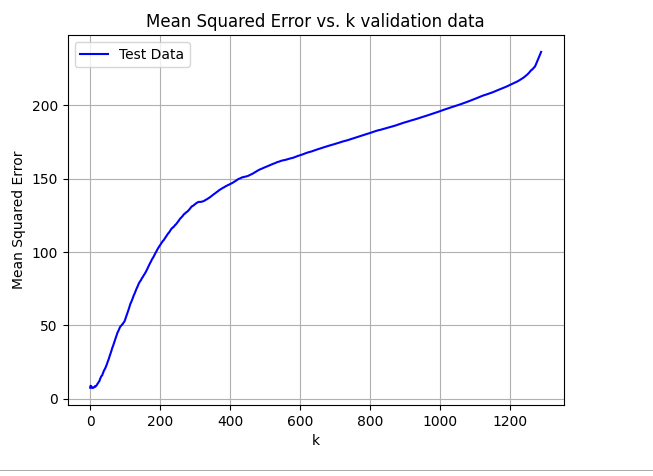
Description automatically generated with low confidence

In this case, if I want to make a model with high accuracy, first three principal components will be enough. With these considerations in mind, the next paragraph will explore the specific details and implications of applying PCA in our machine learning analysis.

Varying the hyperparameter of a k-nearest neighbour regression model (k-nn model)

In the k-nn model, the hyperparameter k determines the number of data points used for predicting unseen data. Since I applied principal component analysis to the original features, the training data consists of principal components 1 to 3, which are split into an 8/2 ratio (8 for training and 2 for testing). To identify a suitable k value for achieving higher accuracy in the model, I tested values from 1 to 1290. The chosen range is based on the total of 1292 instances present in the principal components’ dataset. The results are presented in the figure.

Figure k-nn performance 1-3

 A graph with a red line

Description automatically generated with low confidence

A graph with a blue line

Description automatically generated with low confidence

After conducting 1290 runs with different k values ranging from 1 to 1290, under the condition of random train-test splits (8/2) for each run, the resulting figures show the mean square error for the test data (left) and the training data (right).

On the test data figure, there is an increasing trend with increasing of k in general, but a tiny upward curve appears before k= 50. It seems that sweet spot (minimum mean square error) appears at range 0 to 20. Beyond this range, the line becomes smoother, it means 𝑘-NN is unable to capture the trend in the data as 𝑥 increases. Because the averaging effect will wash out all interesting patterns in the data. And model gets underfit. However, there is overfitting situation on the model because when k is small, training data gets minimum value. This case is different with example graph in the textbook because my figure does not have an obvious upward curve.

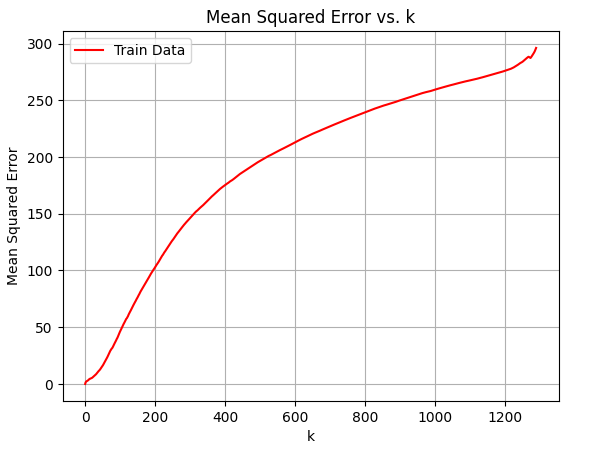
I think there are several reasons. First one is normalization problem, PC1, PC2, PC3 values may have large range, and other has small range, in the end, one value dominates result due to Euclidean distance calculation. it causes the losing information of original data.

Second reason is similarity property of food, two close components and components level of food prefer to have almost identical fat content, which means if k is small, mean square error will also get small on both training and test data.

To test my hypothesis, I decided to normalize inputs using MinMaxScaler in the library sklearn. The figures below are the knn model mean square accuracy which loops 1290 times with different k.

Figure normalization 1-3

A graph with a blue line

Description automatically generated with low confidence

A graph with a blue line

Description automatically generated with low confidence

Left figures are test data accuracy and zooming in, right figure is training data accuracy.

Now we can have evidence indicating that input normalization is dominant the reason of mismatch between textbook and real situation. the nature of data may play a part role in why model does not overfit when k is small. And the best k appears at range (12-16) with mean square error around 9. After finding the reason of novel figure, I want to apply k-fold cross validation on k-nn model to find the best hyperparameter for unseen data since k-fold cross validation can reduce the noise in our model validation and evaluate performance of model on unseen data.

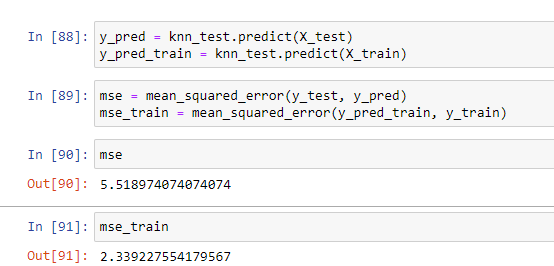
Using cross-validation to validate k-NN models with varied k.

In the k-fold cross validation, I set fold 10 and perform k-nn models with varied k from 1 to 601, and this setting is chosen for two reasons, one is my computer is not affordable the computation higher than that, it can take long time for waiting results, second reason is that lager than 600 k is losing the interesting patterns in the data, which will lead to a meaningless model. figure below shows the result I get with different hyperparameter.

A picture containing text, plot, line, screenshot

Description automatically generated

The figures shows that the best hyperparameter appears at very start In this case, k is 5 with around around 98.7% accuracy. Before applying k fold cross validation, I use train-test split with ratio 8/2, train data is used to do k fold cross validation, the remaining data is used as test set to avoid a risk of overfitting to the validation data. After best k 5 obtained, I train the model with this hyperparameter, and use it to predict data in remaining dataset (1/5). Then mes\_tarin means model is not overfiting to training data, mse means model still get a good predication on unseen data.



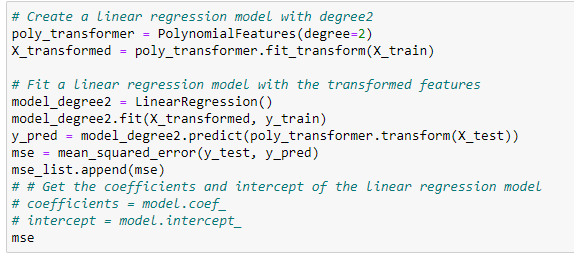
Linear Regression model

Next step is to prepare the linear regression model to compare with knn-model. And The linear regression coefficients are related to the correlations between variables, and these relationships remain the same even after scaling or shifting the variables. Therefore, I do not need to scale my varibles. I decide to do linear regression from degree 1 to 7 (polynomial linear regression), Degree higher than 7 is too complexity and easily overfit to training data, higher degrees are not considered, comparing their performance with put variables PC1, PC2, PC3 and ouput variable [‘fat’]. Then they are applied to do L2 regularisation and find best hyperparameter lambda on unseen data by k fold cross validation.

Among these 6 models, I calculated the mean square error of each models. Result is below

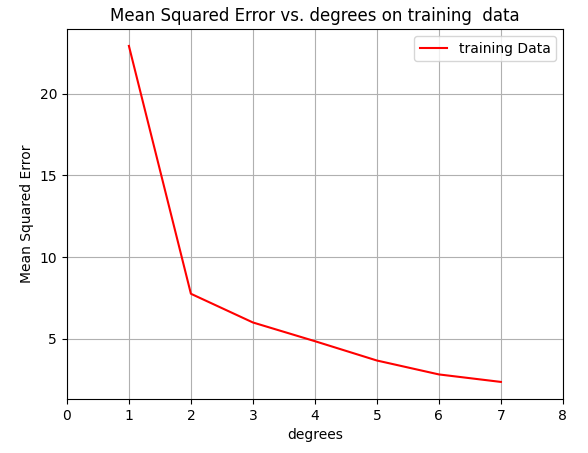
A screen shot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

A graph with a blue line

Description automatically generated with low confidence 

From the above test data graph, with degree increasing, mean square error increases, but mse of training data is decreasing. this decreasing on training data indicates that models are overfitting to training data.

From information of mean square error, linear regression with degree 1 compared with linear regression degree 2-5 has relative bad performance on capturing the relationship between input principal components and ouput variable ‘fat’, it may be that 1st degree model is not suitble for high dimension inputs or it is underfit to capture the pattern of data since it has high mse on both training data and test data. However, models starting from degree 6 to 7 indicates a overfitting the model to the noise rather than the interesting patterns with the increasing degrees of linear regression. Reason may be that high degrees gives model more flexibility and they are overfit the training data. For all models, I would like to perform another machine learning technique, L2 regularization, to mitigate overfitting, then to see whether they have better performance.

L2 regularization

L2 regularization is a machines learning technique which is used to mitigate the overfitting for polynomial regression. It works by adding an extra penalty term, multiplied by a hyperparameter and Euclidean vector nom square of parameter, to squared error loss equation. For finding the best hyperparameter of model, we can use k fold cross validation.

In the next L2 regularisation, I set range of lambda is from 0 to 500 since higher lambda causes model constrained and unbale to capture pattern of data. If some models show an interesting line, I will run more lambda values.

After applying L2 regularisation on degree 1 polynomial regression we have   
A picture containing text, screenshot, line, plot

Description automatically generated

From graph, we can see that mean square error of training data is increasing and line of test data forms a curve with minimum around lambda = 25. Respect to mean square of test data, it experiences a decrease and going up by passing its minimum. The mean square error of non-regulation model original is 22.4 and new minimum is around 21.3, which means that L2 regularization with a certain lambda value gives model more generalization ability. And mean square error of training data constantly increases because of large penalty of lambda which force seta to 0 and model eliminates some important features' coefficients, leading to oversimplified model, which contributes increasing MSE of both training and test data.

After applying L2 regularisation on degree 2, 3, 4 polynomial regression we have

A picture containing text, line, screenshot, diagram

Description automatically generated A picture containing text, screenshot, line, plot

Description automatically generated

A picture containing text, line, screenshot, diagram

Description automatically generated  
on the graph, lines of test data and training data are constantly increasing, and it indicates original models are not overfitting to training data because both test and training data starts from low mean square error value rather than 0 MSE of training data. However, increasing lambda leads to oversimplified model, it means that models from degree 2 to 4 gradually have a performance reduction on predication of unseen data.

After applying L2 regularisation on degree 5 polynomial regression we have an interesting graph:

A picture containing text, screenshot, line, plot

Description automatically generated A picture containing text, screenshot, diagram, line

Description automatically generated

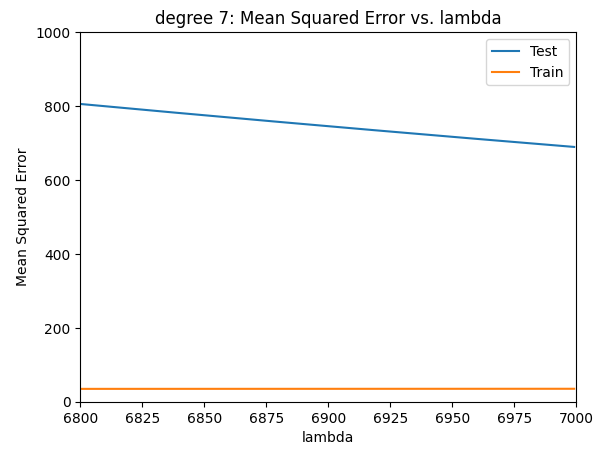
Figures of 1 to 4 have range of lambda from 0 to 500, but X-axis of degree 5 polynomial regression extends to 1000 to obtain clear stable trends for test and train data. As we can see that, when the value of lambda is small (less than 17), the test line starts at a higher mean squared error (MSE) value of 11 and then increases. MSE of training data start from nearly 0. This indicates that the model is overfitting the training data, having bad generality ability. As lambda increases, the regularization effect of Ridge regression becomes stronger, and the model starts to generalize better, resulting in a decrease MSE of test data and an increase MSE of training data. Therefore, it is better to choose lambda equals to 1000 in this range because value does not overfitting to training data and still have a good result on unseen data.

And the same situation also appears to model with degree 7.

A picture containing text, screenshot, line, plot

Description automatically generated A picture containing text, screenshot, line, plot

Description automatically generated



Model is overfitting to training data, we can see from graph, the training line is almost close to y-axis where y = 0. L2 regularization mitigates the overfitting of model and model gets better result on predication of test but model is still overfitting to training data, it is why it test mean square error increases again. But it is still overfitting in the end of lambda range, therefore, this model is meaningless for prediction on unseen data. Last model is degree = 6:

A picture containing text, screenshot, display, number

Description automatically generated A picture containing text, screenshot, line, display

Description automatically generated

At start this model is overfitting to training data and has bad generalization ability. However, the regularization effect starts to work, and the model starts to generalize better, which is reason of decreasing of test line and increasing trend of training line.  
  
next step is to apply k fold cross validation on each model, pre-conditions are lambda in range 0 to 2000 and 10-fold (computer computtaion limiation), dataset is splitted into train and test with 8/2, training data used for k fold corss validation and test data for testing optimal lambda performance on unseen data, after getting optimal lambda, , i then I get optimal degree 2 with MSE 7.2942 and lambda 0, 3 with MSE 4.9353 and lambda 0, 4 with MSE 4.8348 and lambda 3, 5 with MSE 78.6763, 6 with MSE 586.7514 and lambda 52, 7 with MSE 20298.4601 and lambda 2000.

In conclusion, I think the best model in range from 0 to 2000 is redge regression model with degree 4 and lambda 3 because it has best performance on unseen data among models. As for relative bad result of other models, they may constraited by the limited range of lambda, or insufficient data and lack diversity of data.

In summary, data preparation and cleaning are first thing to do, because the quality of data can directly impact the accuracy of final prediction of two models, and dataset is splitted into 8/2 corresponding to training and test data after data preparation. I decided to use PCA on data to improve the predication accuracy of two models. Then, normalization is used to scale input variables and cross vlidation mtheds is used to obtain best k (5) on unseen data.

For linear regression model, I decided to to increase its complexity by rising degrees from 2 to 6 of model, L2 regualation is applied on each model to mitigate overfitting of high degree models. Best lambda value of each model is obtained by k fold cross validation. By comparing their performance on test data. Model with degree 4 and lambda value 3 has best performance (mse 4.8348).

From the mean square number of two best models, ridge regresion model has better performance on unseen data in the conclusion respect to trained on this dataset.

for this dataset The accuracy difference between two models have several reasons , one reason is that diveristy of data. Figures of knn-model indicates that models are get more accuracy on lower k value, this maybe a sign of lack of diveristy on training dataset. Solution is simply to add more data to balance the original data. Another one maybe insufficient training data, this dataset only have 1616 instances available, which may contributes to increasing difficulty of capturing the relationship of data for knn model.

As for ridge model, the result of mean square error is better. However, It can have better performance by incrasing data diveristy and adding more data into dataset.

Appendix

Figure1

0 value occupation for each column

A picture containing text, screenshot, font, number

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

Checking missing value for each column after dropping columns

A screenshot of a computer

Description automatically generated with low confidence

Correlation

A picture containing text, screenshot, diagram, line

Description automatically generated

Scatter plot

