Bitbucket Link : git clone git@bitbucket.org:ckochh2/hw5.git
Write Up: TASK: Option 2: How likely a person is to "pay more money for good, quality or healthy food" (on a scale from 1 to 5).

ML Solution Used: With Hyper parameter Tuning (depth =5), Random Forest gave the best test accuracy i.e. 45% with precision and recall values to be 51% and 45% respectively. On trying the same with SVM (rbf kernel), it gave a test accuracy of 44%.
Reason: Random forests can be used to rank the importance of variables in a regression or classification problem in a natural way. The first step in measuring the variable importance in a data set D is to fit a random forest to the data. During the fitting process the out-of-bag error for each data point is recorded and averaged over the forest. Since the target variable (Spending on Healthy Eating) is dependent on important feature and random forest split on imp. features first the model is able to predict relevant test accuracy. It doesn't over fit also as the model is trained with cross validation multiple times with shuffling of the dataset, the accuracy avg remains the same.
Reason to use SVM : SVM kernel can help fit data which are not linearly separable. Here rbf kernel was able to give 44% accuracy on test data.
Pre-Processing Data :

- Handled Categorical data : Used Get_dummies() function which converts categorical variables into dummy/indicator variables for categories like : ['Gender','Left - right handed','Only child','Village - town','House - block of flats']
- Handled Missing Values : With the most frequent values or mean.
- Continuous values are divided into bins of 5. For columns Age, Height, Weight.
- Since all values are scaled from 1 to 5, the data gets normalized and thus dropping features or rows is not needed as the outliers were less and were normalized by scaling and for the most important features the sum of null values was hardly above 20.

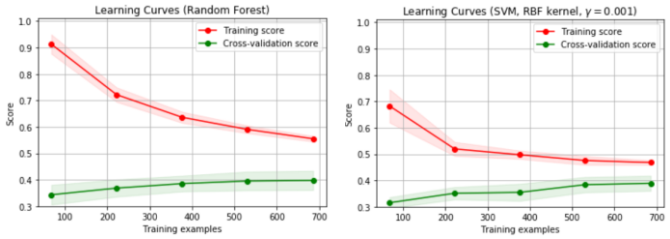
Feature Extraction: Correlation matrix with respect to target variable is computed based on which the most important features are selected. Used scikit learn's sklearn.feature_selection import SelectKBest to extract best features but correlation matrix gave a better accuracy.

Software Used: Numpy, Pandas, Scikit Learn Library and Ensemble Method, Cross Validation, GridSearchCV, GuassianNB, RandomForest etc.

Evaluate Success: The success of the model can be evaluated in terms of accuracy achieved, the precision and recall values which are 45%,51% and 45% respectively.

1. The selection of best model can be determined by scikit learn GridSearchCV which tune the best hyperparameter and return the results over cross validation.
2. We can use ensemble methods like GradientBoostingClassifier supports both binary and multi-class classification. It fits the data with 'n' no. of decision stumps as weak learners and controlled either by setting the tree depth or number of leaf nodes. The learning rate is a hyper-parameter in the range (0.0, 1.0] that controls overfitting
3. Random Forest is intrinsically suited for multiclass problems than SVM. Random Forest works well with a mixture of numerical and categorical features. In a many class case especially if we expect to have outliers, then using Random Forest can give better result (using subset of training sets with bagging and subsets of features can help reduce their effect)

Results: Learning Curve : With the increase in no. of samples the test accuracy increase as there is more data to analyze on. On small dataset the model overfits.



Bitbucket Link : git clone git@bitbucket.org:ckochh2/hw5.git
Write Up: TASK: Option 2: How likely a person is to "pay more money for good, quality or healthy food" (on a scale from 1 to 5).