Question3:

a) **Decision Tree Implementation(using python):**

No.of folds =1 ; max_Depth is to limit the height of the tree. And min_size is the minimum number of rows a node can have. Using the training data, decision tree is built by calling build_tree. Best_split gives the best possible split. Initially, it finds the attribute(column) based on which partition can take place. For each attribute, it finds all the unique values. Then for each unique value, it splits the data into two groups and then entropy is calculated by **check_entropy.** As reference, parent's entropy(e_val) is taken to be 1. **Information gain = e_val - return(check_entropy)** . For each attribute, for each value in the attribute's column, information gain is calculated and the one that gives highest information gain is stored in respective variables. After this step, attribute and value of the attribute is knows. Split() function then basically constructs a tree. The nodes maybe empty(i.e, pure set), depth might exceed max_depth and size of each node maybe less than min_size. All these cases nodes are made terminals and represented by the truth value that is repeated most number of times. If they are not terminals, then nodes are further split(by calling best_split followed by split function). By end of this tree is built using training set. Now test_set is sent. Predict() function is used to predict the truth value that the particular row in the test_set would give on passing it to the tree. If the predicted value and actual value(last column) are same then true is inserted into result list. Thus for a test_set, result is list of trues and falses.

For following parameters,
N_folds =1
Max_depth =5
Min_size =1

Resulted accuracy = 0.815950920245

b) **Using Cross-validation:**

In this case number of folds=10; The given partition function for testing and training set in the given skeleton code is changed. Q2b.py is the file with 10 cross validations.
On changing only the number of folds;

Resulted accuracy = 0.80542923918

c) **Improvement Strategies:**

On increasing the max_depth to 20 in case(a), accuracy is increased to 0.856850715746
On using gini method, accuracy = 0.783231083845

Question 4:

Method1: Using MultinomialNB
Score = 0.74074
Method2: Using ComplementNB
Score = 0.71611

In both cases, tf-idf method is used. Frequency of each ingredient is stored in a matrix. Then this matrix and the cuisines(training_data's) list is sent to different classifiers and then testing_data is sent into classifier's predict function.

On using kNN with k=10, the score was considerably low. Because of high data and overlapping of classes, this must have occured.