Random Forest and KNN classifiers

In this assignment Wisconsin breast cancer data set was used to train a Random forest and KNN classifiers using python's *sklearn.ensemble.RandomForestClassifier* and *sklearn.neighbors.KNeighborsClassifier* libraries. First the data set was loaded into a pandas data frame and observed to be consisting of 10 features and a class label representing whether each data point represents a benign or malignant breast cancer. Since original data set did not have feature names assigned in the columns, appropriate names were given to each of the 11 columns including the class label.

It was observed that data set contained some missing values on '*Bare_Nuclei*' feature represented as '?' value. Initially the missing values were replaced with null (np.nan) and then python's univariate imputation library was used to replace the null values with the most frequent value of each dimension. After imputation, data types of the columns were observed to have changed to type *object*, and they were converted back to original data type that was *int64*. After handling missing values, the feature representing '*sample code number'* was removed from the data set as it did not represent any real feature but a code number given to each sample. *sklearn.model_selection.train_test_split* was used to split the original data into training and test sets using the same ratio as done in earlier assignments that was 35%.

For random forest classification, *sklearn.ensemble.RandomForestClassifier* was used with default values and trained with above split training data and predicted on test data set each having 9 features. *sklearn.metrics.classification_report* was used to generate a classification report comparing the predictions and the actual results providing precision, recall, f1 score, and support values for benign class as (0.95, 0.97 , 0.96, 160) and for malignant class as (0.94, 0.90,0.92, 84). *Confusion_matrix*, and *accuracy_score* methods generated values as 76 true positives, 155 true negatives, 5 false positives, 8 false negatives and overall accuracy score of the model to be 95.08%. Then the classifiers *feature_importances_* method was used to generate the importance score for each feature to the classification and it was identified that '*Mitoses'* feature was having a very low importance to the model. Since such features can introduce noise to the classifier, above process was re followed after removing '*Mitoses'* feature from the data set. As expected, now the classifier gave an improved classification report as below.

	Precision	recall	F1 score	support
Benign	0.96	0.98	0.97	167
Malignant	0.96	0.91	0.93	77

The confusion matrix gave results of 70 true positives, 164 true negatives, 3 false positives, 7 false negatives. Overall accuracy of the model was now improved further to 95.9% and this was the highest accuracy obtained for this data set so far when compared with logistic regression (93%) and C4.5 (90%).

For KNN classification same cleaned data set above without 'sample code number' feature was used (9 features in total) with *sklearn.neighbors.KNeighborsClassifier* classifier. In order to find the best k value for the data set, 50 instances of KNN classifier were trained with the same training to test split recording the error of the classifier in each case and observed that k=11 gave one of the minimal error rates and the classifier is re run setting k=11 by recording the classification report and confusion matrix as below.

	Precision	recall	F1 score	support
Benign	0.98	0.98	0.98	160
Malignant	0.96	0.96	0.96	84

Results of the confusion matrix were 81 true positives, 157 true negatives, 3 false positives, 3 false negatives. And KNN has shown the highest overall accuracy score of 97.5%.