REPORT-Plastic Classification

~Hetvi Patel

Analyzing the data:

XPlastic: This 2D numpy array was 500x36 in shape i.e., it contained the values of absorbances at 36 different wavelengths for 500 samples. The minimum absorbance in the array was 0.0 (which means the transmittance is 100%) and maximum value is 5112.85.

YPlastic: This 1D numpy array had the names of plastic that each of the samples in XPlastic belonged to. There are 5 types of plastic: Plastic 1, Plastic 2, Plastic 4, PVC and White Plastic in the numpy array. A special parameter of the numpy, allow_pickle was used to read the YPlastic array since, it contained object elements which can only be read only when allow_pickle is set to True.

Classifiers Used:

1. **KNN Classifier**:
   It is based on Supervised Learning technique. It assumes similarity the new data point and puts it into the category that is the most similar to the categories available. KNN is used for classification as well as regression. However, in most cases it is used for classification purpose.

   There are different parameters on which KNN classifier depends upon. Out of all the parameters only the parameter n_neighbors was considered which was set to 5. n_neighbors specifies the number of datapoints to be considered to decide the category of the new data point. Keeping n_neighbors to 1 gives the maximum accuracy but in that case, there are chances of overfitting. Therefore, after observing the accuracy at different values, the accuracy was maximum at n_neighbors=5. Other parameters are set to their default values. Their values can also be changed but it was observed that the accuracy had no change in changing any of the other parameters. Thus, it was concluded that hyperparameter tuning was of no use here.

   The maximum accuracy obtained was **0.9133** (91.33%). accuracy_score is used to find the efficiency of classification models whereas r2_score is used for regression model. The model made is for classification, therefore, accuracy_score is used to find the efficiency.

2. **Decision Tree Classifier**:
   It is also based on Supervised Learning technique. It has two nodes at each step, the Decision Node and the Leaf Node. The Decision nodes are used to make any decision and have further branches whereas Leaf nodes depict the final output of the decision and do not contain further branches. Decision Trees are used for both Regression problems and Classification problems, however most of the use is in classification.

There are many parameters that the classifier depends on out of which criterion, splitter, max_depth and random_state were taken into consideration whereas default values were automatically used for the parameters not specified. For the parameters specified, hyperparameter tuning was done to get maximum accuracy. random_state was set to 0 to avoid randomness of the estimator.

accuracy_score was used to find the accuracy and the maximum obtained value was **0.9867** (98.67%)

3. **Gradient Boosted Decision Tree (GBDT) Classifier**:
   It gives a prediction model which is in the form of an ensemble of weak prediction models which are basically decision trees. When the decision trees are weak, GBDT are used. It uses stage-wise fashion similar to the other boosting methods, however GBDT generalizes the other methods by allowing optimization of the differentiable loss function.

   The random_state was set to 0 to avoid randomness of the estimator.

   The accuracy obtained was **0.98** (98%).

4. **Linear Support Vector Classification (SVC)**:
   Linear Support Vector Machine is a Supervised Learning Algorithm which is mostly used for classification but it can also be used for regression. The algorithm tries to find the optimal hyperplane that can be used to classify new datapoints using the training data. Normally, learning algorithms tries to learn the most common characteristics of a class and classify the rest based on the differences. However, SVM, finds the most similar examples between classes (known as support vectors).

   The different parameters considered are multi_class, C and random_state. random_state is set to 45 to avoid randomness of estimator and the value of C is set at 10. Nonetheless, changing the value of C does not affect the accuracy much.

   The accuracy obtained after tuning the parameters was **0.9667** (96.67%).

5. **Kernelized Support Vector Classification (Kernelized SVC)**:
   Kernelized Support Vector Machines are used when normal SVM is unable to separate the dataset in the current dimensions and there is a need to transform to higher dimensions.

   The parameters kernel, degree, gamma, decision_function_shape and C were passed into the classifier. Using hyperparameter tuning, the best set of parameters that could give the most accurate model was considered.

   The maximum accuracy obtained was **0.9534** (95.34%)

6. **Naive Bayes Classification**:
   Naive Bayes classifiers are a collection of classification algorithms based on the Bayes Theorem. There is no single algorithm but a set of algorithms that share a common principle. Thus, every pair of features that is classified is independent of other pairs.

   The accuracy using Naive Bayes Classification was **0.9734** (97.34%)

7. **Random Forest Classification**:
   This Classification algorithm is also based on the Supervised Learning Algorithm. Like many other algorithms Random Forest can also be used for both classification and regression. It is based on the concept of ensemble learning. The random forest classifier is same as the gradient boosted classifier but there are some differences. The most significant among them is Random Forest Classification is used as a parallel estimator where each tree is fit to a subsample taken from the entire dataset whereas in the Gradient Boosted Decision Tree classification, the trees are connected sequentially to obtain a strong learner. Also the decision tree in GBDT are not fit to the entire dataset.

   The parameters considered are criterion, max_depth, bootstrap and random_state. bootstrap specifies whether bootstrap samples should be used to build trees or the entire dataset should be used. Making bootstrap=False, picks the latter option. random_state was set to 0 to avoid random initialization of the estimator. To pick the best set of parameters, hyperparameter tuning was done.

   The maximum accuracy obtained was **0.9933** (99.33%)

Results:

Out of all the model, the maximum accuracy obtained was **0.9933 (99.33%)** using the **Random Forest Classifier**.