A2: GIVE YOUR MODELS A GRADE

AI 539 ML CHALLENGES

SUBMITTED BY : PUSHPAK VIJAY KATKHEDE

1) Classifiers used in the assignment –
We have used 4 classifiers as follows :

1. **Decision Tree** – In decision tree the algorithm generates a root node which is most important attribute. For evaluation we start at the root node and work our way down the tree by following the corresponding node that meets our condition or "decision". This process continues until a leaf node is reached, which contains the prediction or the outcome of the decision tree.
Hyper parameter settings –
DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=9, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0)

2. Random Forest – The core unit of random forest classifiers is the decision tree. the bootstrapping technique helps the development of random forest with a set of required number of decision trees to improve classification accuracy.

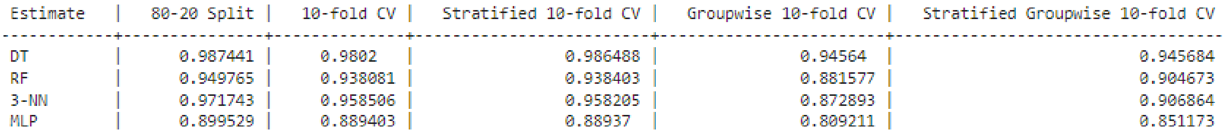
Hyper parameter settings –
RandomForestClassifier(n_estimators=100, *, criterion='gini', max_depth=2, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=9, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None

3. K Nearest Neighbor – KNeighborsClassifier is a supervised learning algorithm that makes ` classifications based on data neighbors. With KNN we can have a certain set of data and from it draw patterns that can classify or group our data.
Hyper parameter settings –
KNeighborsClassifier(n_neighbors=3, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None)

4. MLP Classifier – A multilayer perceptron (MLP) is a fully connected class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to mean *any* feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation); see § Terminology. Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden lay (from Wikipedia)

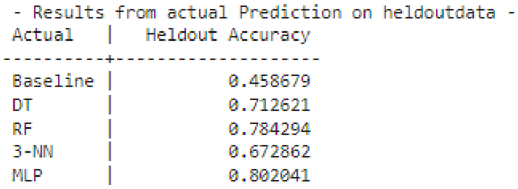
Hyper parameter settings –
MLPClassifier(hidden_layer_sizes=(100,), activation='relu', *, solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=300, shuffle=True, random_state=9, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000)

2) Classification accuracy (generalization estimate) for each evaluation methodology –



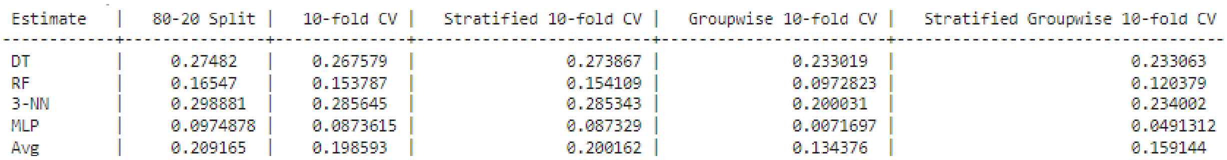
```
 Estimate    |   80-20 Split |   10-fold CV |   Stratified 10-fold CV |   Groupwise 10-fold CV |   Stratified Groupwise 10-fold CV
-------------+---------------+--------------+-------------------------+------------------------+-----------------------------------
 DT          |      0.987441 |     0.9802   |                0.986488 |            0.94564     |                          0.945684
 RF          |      0.949765 |     0.938081 |                0.938403 |            0.881577    |                          0.904673
 3-NN        |      0.971743 |     0.958506 |                0.958205 |            0.872893    |                          0.906864
 MLP         |      0.899529 |     0.889403 |                0.88937  |            0.809211    |                          0.851173
```

3) A) Held-out (actual) classification accuracy for each classifier and baseline classifier-



```
 - Results from actual Prediction on heldoutdata -
 Actual    |   Heldout Accuracy
-----------+--------------------
 Baseline  |           0.458679
 DT        |           0.712621
 RF        |           0.784294
 3-NN      |           0.672862
 MLP       |           0.802041
```

B) Signed error between each method's generalization estimate and the actual classifier performance –



```
 Estimate    |   80-20 Split |   10-fold CV |   Stratified 10-fold CV |   Groupwise 10-fold CV |   Stratified Groupwise 10-fold CV
-------------+---------------+--------------+-------------------------+------------------------+-----------------------------------
 DT          |     0.27482   |    0.267579  |                0.273867 |             0.233019   |                          0.233063
 RF          |     0.16547   |    0.153787  |                0.154109 |             0.0972823  |                          0.120379
 3-NN        |     0.298881  |    0.285645  |                0.285343 |             0.200031   |                          0.234002
 MLP         |     0.0974878 |    0.0873615 |                0.087329 |             0.0071697  |                          0.0491312
 Avg         |     0.209165  |    0.198593  |                0.200162 |             0.134376   |                          0.159144
```

4) Answers-

A. Using the development data only (Step 4), for each evaluation methodology (A-E), which classifier was expected to perform best in the future?

→ For every evaluation methodology the expected best performing classifier in future as below :

- ➢ Train-test split (train = 80%, test=20%) - Decision Tree (accuracy - 0.987)
- ➢ 10-fold cross validation – Decision Tree (accuracy - 0.9802)
- ➢ Class-stratified cross validation - Decision Tree (accuracy - 0.986)
- ➢ Group-wise cross validation - Decision Tree (accuracy - 0.945)
- ➢ Class-stratified group-wise cross validation - Decision Tree (accuracy - 0.945)

B. Which classifier performed best on the held-out test set (step 5A)?

→ MLP performed best with accuracy of 0.802 on the held-out data.

C. Which classifier(s) performed better than the baseline on the held-out set?

→ All classifiers (Decision Tree, Random Forest, KNearestNeighbour, MLP) performed better than the baseline on the held-out set.

D. Which evaluation method had the smallest error (averaged across classifiers) between estimated test performance (from the development data) and actual held-out performance (step 5B)?

→ Groupwise 10-fold has the lowest averaged error (0.134) across all the classifiers among all methods.

E. Why do you think there is a difference in the generalization estimates made by methods B, C, and D on this data set?

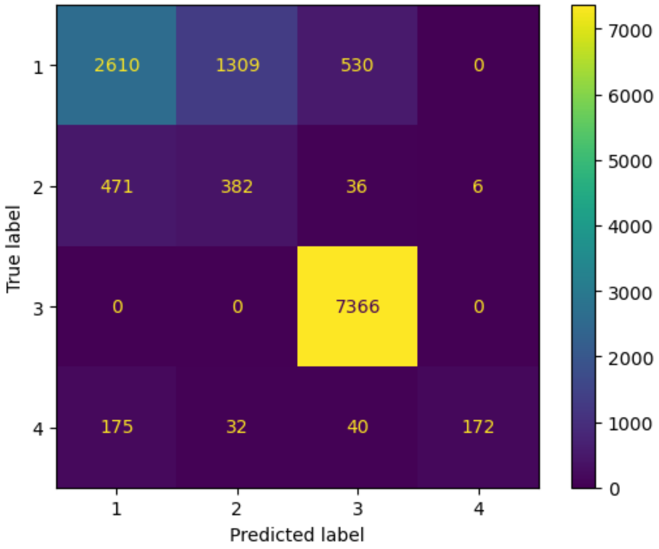
➔ In every evaluation method, the folds generated does have different data points that are included as per their particular strategy. In B, the folds do follow any constraint with respect to class or group of the person. Whereas, in C the folds are made such that the percentage of samples for each class is preserved in each fold with respect to that of the original set. In D, the folds are balanced in such a way that number of distinct groups is approximately same in each group. Hence, all this leads to generation of different data in any respective fold compared across B,C and D horizontally and eventually which leads to the difference in generalization estimates of these methods.

F. The held-out set contained observations from 10 new people. If instead we have a test set that contains new observations from the same people from the training set, which evaluation methodology (A-E) would give the best generalization estimate (smallest gap between predicted and actual accuracy), and why?

➔ If the new observations are collected from the same people then most of the features will remain same or very close to the training set. So, the method which co relates to the selection of a specific person will be more accurate and the feature ''person" will be exploited more. Hence, I think the method D i.e., Group Kfold will give us the best generalization estimate.

G. Show the 4x4 confusion matrix for the classifier that performed best on the held-out set. Which kind of error do you think is the most important to avoid/penalize?

➔ The confusion matrix generated is as below:



I think the errors where the classifier wrongly predicted class 1 when it is another should be important to avoid. These errors constituted the highest number of errors mostly half of the total errored predictions. Hence, dealing this error with cost sensitive learning may be very helpful to significantly improve the performance of the classifier.

This confusion matrix is generated over predictions from the best performed classifier (MLP classifier) on the held-out data. Generated by the program in the file confusion_matrix.py separately and will provide same accuracy as in the main program file as the parameters used are the same in both here and main program in activity_eval.py.

H. Given what you know of how the data was collected, what kinds of bias may be present in this data set? (Recall Mehrabi et al. (2021), "A Survey on Bias and Fairness in Machine Learning", https://dl.acm.org/doi/abs/10.1145/3457607)

➔

A) Omitted variable bias – believe there must be other factor that may affect the observations collected by the accelerometer like if the height of bed is more than its usual height it may call the person to fall in ambulating category instead of him being sitting on the bed. Also, other factor like wind may affect the observation of the accelerometer.

B) Representation Bias – Also, the class 3 (lying down) is comparatively highly represented hence the data is not properly generalized across all the classes.

C) Longitudinal Data Fallacy – The data with time change as the features like 1,2,3 will change with time.

I. Reflection: What did you learn from completing this assignment? How long did it take (in hours)? What was the hardest part? What might you use in the future?

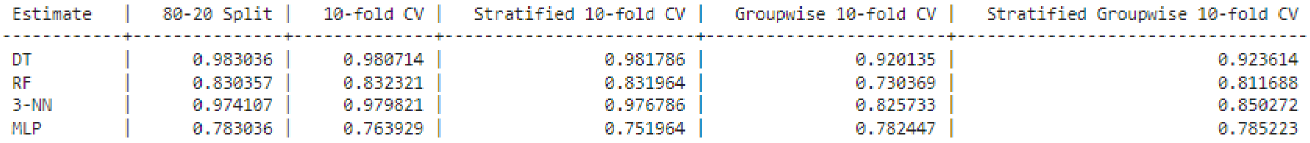
➔ I learnt one important thing that the classifier that initially appeared best may or may not perform the best over the predictions. Hence, it is better to validate the classifiers with possible evaluation strategies. In total it took 6-7 hours for me including extra credit question and documentation. Hardest part was nothing as such but had faced an error with training that took a little more time to get resolved. The code generated is going to be helpful in my project. Also, the reading that I did about the hyper parameters of all the classifiers helped me understand more about the specifics of each of the classifiers and generalization metrics of each of them.

5) **Extra Credit –**

Here, there was high imbalance in set favoring the class 3 almost 10x times the other classes. Here, to deal with this we oversampled the data using SMOTE and reduced the observations of the majority class to 50%. And, oversampled the minority class 5 times to generated equivalence in the representation of each of the classes in dev2 set.

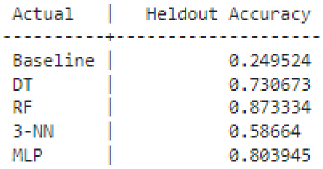
Find the observations generated by activity_eval_for_dev2.py file which deals with the newly generated dev2 training set which is dealt with class imbalance through oversampling and under sampling the observations.

Estimation of accuracies through A-E methods of CV –



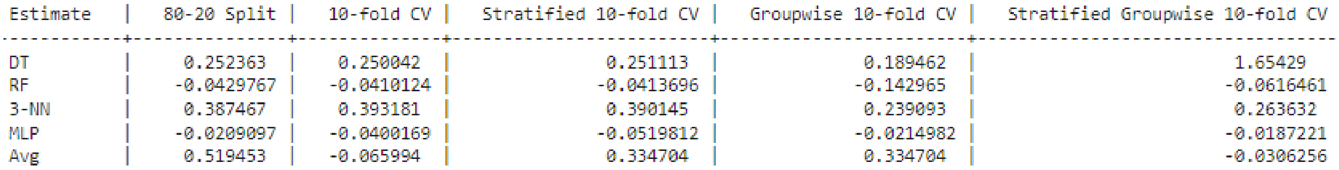
```
 Estimate    |  80-20 Split |  10-fold CV |  Stratified 10-fold CV |  Groupwise 10-fold CV |  Stratified Groupwise 10-fold CV
-------------+--------------+-------------+------------------------+-----------------------+----------------------------------
 DT          |     0.983036 |    0.980714 |               0.981786 |              0.920135 |                         0.923614
 RF          |     0.830357 |    0.832321 |               0.831964 |              0.730369 |                         0.811688
 3-NN        |     0.974107 |    0.979821 |               0.976786 |              0.825733 |                         0.850272
 MLP         |     0.783036 |    0.763929 |               0.751964 |              0.782447 |                         0.785223
```

Results from actual predictions on held-out data –



```
 Actual    |  Heldout Accuracy
-----------+-------------------
 Baseline  |          0.249524
 DT        |          0.730673
 RF        |          0.873334
 3-NN      |           0.58664
 MLP       |          0.803945
```

Error report between estimated and actual predictions across all CV methods.



```
Estimate    |  80-20 Split |  10-fold CV |  Stratified 10-fold CV |  Groupwise 10-fold CV |  Stratified Groupwise 10-fold CV
------------+--------------+-------------+------------------------+-----------------------+----------------------------------
DT          |    0.252363  |   0.250042  |               0.251113 |              0.189462 |                          1.65429
RF          |  -0.0429767  |  -0.0410124 |             -0.0413696 |             -0.142965 |                       -0.0616461
3-NN        |    0.387467  |   0.393181  |               0.390145 |              0.239093 |                         0.263632
MLP         |  -0.0209097  |  -0.0400169 |             -0.0519812 |            -0.0214982 |                       -0.0187221
Avg         |    0.519453  |  -0.065994  |               0.334704 |              0.334704 |                       -0.0306256
```

a) This time the Stratified Groupwise 10-fold has the lowest generalization error (0.0306) according to the table.

b) Here, as the class imbalance has been dealt with the previous process only it has been a better push towards E method hence, I think method E will be better evaluation methodology for this problem.