Dear Dr. Hazem, Dr. Mahmoud,

After receiving the extracted features from Dr. Philipp Werner, we created a new folder containing the following:

- “Table\_Step2\_159Features-85Subs-5Levels-z.xlsx” – the sent file containing the features.  
-“ANN.py”: classification using artificial neural networks  
- “DecisionTreeClassifier.py”: classification using decision.  
- “KNN.py”: classification using K-Nearest Neighbor.  
- “NaiveBayes.py”: classification using naïve bayes.  
- “RandomForest.py”: classification using rand  
- “SVM.py”: classification using artificial neural networks  
- “SVM\_Linear.py”: classification using artificial neural networks  
- “SVM\_RBF.py”: classification using artificial neural networks

Notes: Describe the parameters used for the classifier”

In each file we’ve used different new model for classification, following the same procedures:  
1- Importing the dataset using the pandas library.  
2- Excluding the first column that contains the label, and imputing the missing values for the rest of the columns by replacing it with the mean value of the column.  
3- Separating the whole dataset into 5 subsets for each level for the 5 levels.  
4- Design the learning mechanism:

* + 1. Preparing the subsets that will be fed to the sequent, by combining the subsets.
    2. Splitting the dataset into training and testing set with 75% - 25%.
    3. Using the training set for the learning.
    4. Calculating the evaluation metrics.

We’ve summarized the results as the following:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | B vs. T1 | B vs. T2 | B vs. T3 | B vs. T4 | B vs. T1 vs. T4 | B vs. T1 vs. T2 vs. T3 vs. T4 |
| 1. Artificial Neural Networks: | | | | | | |
| Confusion Matrix |  |  |  |  |  |  |
| Accuracy | 0.769411764706 | 0.802352941176 | 0.842352941176 | 0.8576470 |  |  |
| Precision | 0.77829099 | 0.80327869 | 0.83066362 | 0.81458333 |  |  |
| Recall | 0.79669031 | 0.8108747 | 0.85815603 | 0.92434988 |  |  |
| Sensitivity | 0.77304964539 | 0.8297872340 | 0.85815602 | 0.92434988 |  |  |
| Specificity | 0.765807962529 | 0.775175644 | 0.82669789 | 0.79156908 |  |  |
| 1. Decision Tree Classifier: | | | | | | |
| Confusion Matrix |  |  |  |  |  |  |
| Accuracy | 0.7317 | 0.7329 | 0.8188 | 0.8341 | 0.4047 | 0.1816 |
| Precision | 0.7181 | 0.7258 | 0.8149 | 0.8472 | 0.7135 | 0.5975 |
| Recall | 0.7588 | 0.7446 | 0.8226 | 0.8132 | 0.6658 | 0.5737 |
| Sensitivity | 0.7588 | 0.7446 | 0.8226 | 0.8132 |  |  |
| Specificity | 0.7049 | 0.7213 | 0.8149 | 0.8548 |  |  |
| 1. K-Nearest Neighbor: | | | | | | |
| Confusion Matrix |  |  |  |  |  |  |
| Accuracy | 0.6047 | 0.6423 | 0.7117 | 0.8117 | 0.3796 | 0.1684 |
| Precision | 0.6053 | 0.6325 | 0.6926 | 0.7745 | 0.5559 | 0.4395 |
| Recall | 0.5910 | 0.6713 | 0.7565 | 0.8770 | 0.6635 | 0.5105 |
| Sensitivity | 0.5910 | 0.6713 | 0.7565 | 0.8770 |  |  |
| Specificity | 0.6182 | 0.6135 | 0.6674 | 0.7470 |  |  |
| 1. Naïve Bayes: | | | | | | |
| Confusion Matrix |  |  |  |  |  |  |
| Accuracy | 0.5894 | 0.7105 | 0.6847 | 0.7458 | 0.4188 | 0.1849 |
| Precision | 0.5569 | 0.6571 | 0.6240 | 0.6838 | 0.5833 | 0.3402 |
| Recall | 0.8557 | 0.8747 | 0.9219 | 0.9101 | 0.7741 | 0.8009 |
| Sensitivity | 0.8557 | 0.8747 | 0.9219 | 0.9101 |  |  |
| Specificity | 0.3255 | 0.5480 | 0.4496 | 0.5831 |  |  |
| 1. Random Forest: | | | | | | |
| Confusion Matrix |  |  |  |  |  |  |
| Accuracy | 0.7647 | 0.7588 | 0.8435 | 0.8623 | 0.4313 | 0.1929 |
| Precision | 0.7397 | 0.7206 | 0.8020 | 0.8187 | 0.6967 | 0.5362 |
| Recall | 0.8132 | 0.8416 | 0.9101 | 0.9290 | 0.7465 | 0.6932 |
| Sensitivity | 0.8132 | 0.8416 | 0.9101 | 0.9290 |  |  |
| Specificity | 0.7166 | 0.6768 | 0.7775 | 0.7962 |  |  |
| 1. Support Vector Machines: | | | | | | |
| Confusion Matrix |  |  |  |  |  |  |
| Accuracy | 0.7976 | 0.8329 | 0.8670 | 0.9011 | 0.4823 | 0.2169 |
| Precision | 0.7980 | 0.8290 | 0.8587 | 0.8878 | 0.7962 | 0.6659 |
| Recall | 0.7943 | 0.8368 | 0.8770 | 0.9172 | 0.7926 | 0.7658 |
| Sensitivity | 0.7943 | 0.8368 | 0.8770 | 0.9172 |  |  |
| Specificity | 0.8009 | 0.8290 | 0.8571 | 0.8852 |  |  |
| 1. Support Vector Machines with RBF kernels: | | | | | | |
| Confusion Matrix |  |  |  |  |  |  |
| Accuracy | 0.7788 | 0.8141 | 0.8364 | 0.8952 | 0.4690 | 0.2108 |
| Precision | 0.7845 | 0.8018 | 0.8169 | 0.8599 | 0.7880 | 0.6552 |
| Recall | 0.7659 | 0.8321 | 0.8652 | 0.9432 | 0.7281 | 0.7166 |
| Sensitivity | 0.7659 | 0.8321 | 0.8652 | 0.9432 |  |  |
| Specificity | 0.7915 | 0.7962 | 0.8079 | 0.8477 |  |  |
| 1. Support Vector Machines with Linear Kernels an PCA | | | | | | |
| Confusion Matrix |  |  |  |  |  |  |
| Accuracy | 0.8094117 | 0.83058 | 0.86470 | 0.9094 |  |  |
| Precision | 0.8190709 | 0.816326 | 0.8564 | 0.8876 |  |  |
| Recall | 0.79196217 | 0.8510 | 0.8747 | 0.9338 |  |  |
| Sensitivity | 0.7919621 | 0.8510 | 0.8747 | 0.9338 |  |  |
| Specificity | 0.826697 | 0.810304 | 0.8548 | 0.8829 |  |  |
| Cramérs |  | 0.659475 | 0.727243 | 0.81525 |  |  |

“Notes : Fill out the measures for each model.”

Here’s a graphical representation for each metric used, and comparison between the models used for classification:

Here’s a graph showing the performance for each model for each comparison: