MATHEMATICAL PATTERN RECOGNITION

HAND POSTURES DATASET

ABSTRACT

This project is about a pattern recognition problem that predicts the hand posture based on the positions of the markers on the glove. The Hand Posture Dataset is used for this problem statement. The, datasets are standardized, and the preprocessed data is passed through the cross-validation unit to get the best hyperparameters for a specific classifier. Cross Validation technique is used to test the effectiveness of any machine learning model. Sometimes, we have noticed that even tough the model has acquired the desired accuracy, it is not necessary that the model performs well on unseen data. Therefore, cross validation keeps a portion of the dataset aside from D_train and keeps it only for validation (Training is not done over Validation set) and the remaining dataset is used for Training the model. LeaveOneGroupOut cross validation technique sets aside the records of one user for validation and training is done on the remaining users. The cross validation is doe on four Classifiers Naïve Bayes, KNN Classifier, SVM Classifier and Random Forest Classifier. The best accuracy was attained for SVM Classifier with training accuracy of 99.65% and test accuracy of 95.08%.

INTRODUCTION

The Hand Postures Dataset is used for the pattern recognition problem to classify the hand the posture based on the positions of the markers on the hand gloves. The dataset consists of 12 users, and the train dataset 'D_train' consists of all the users expect 3,4 and 7^{th} user. Each of the user consists of 1500 records. Thus, D_train has 13500 records totally. The testing dataset D_Test consists of 3 users and the total no of records are 21099. There are 5 types of hand posture. The X, Y, Z coordinates of each marker is recorded. The details about which finger of the glove has a specific marker is unknown. Thus, preprocessing is essential to convert it to a usuable form. One user can contribute to more data points compared to other users. Thus, LeaveOneGroupOut cross validation should be implemented. This problem statement should be dealt as a user independent posture recognition problem. The data from one user should not be shared amongst the validation, training, and the test datasets. So, the data is split by user ID, and then stratified based on each class. For cross validation only the D_train is taken for consideration. After the best parameters are estimated through the cross-validation step, the same best parameters are applied to the D_Test and the test accuracy is estimated. The goal of the project is to preprocess the data to a usable form and apply cross validation on it for different classifiers.

FEATURE ENGINEERING

- **Generation of 13 new features**

  The data would not work properly in the original form because the no of features for each datapoint and for each class is different. The columns of the dataframe are Class, User, X0,Y0,Z0,X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4,X5,Y5,Z5,X6,Y6,Z6,X7,Y7,Z7,……X11,Y11,Z11.The index like 0 in X0,Y0,Z0 does not give any information regarding the position of the marker. Thus, the data is raw form is not significant. There are a lot of nan values in the original dataset. Thus, the 13 new features that are generated also deals with the nan values. Thirteen new features are derived from the raw data like no of markers, mean of X, mean of Y, mean of Z, standard deviation of X, standard deviation of Y, standard deviation of Z, maximum of X, maximum of Y, maximum of Z, minimum of X, minimum of Y and minimum of Z of marker locations. Thirteen new features are generated.

- **Standardization**
  The train set is standardized by subtracting the mean of specific features and dividing it by standard deviation of those specific features. The resulting data will have zero mean and unit variance.

$$z = \frac{X - \mu}{\sigma}$$

-

FEATURE DIMENSIONALITY ADJUSTMENT

- **Expanding the dataset by adding nonlinear features**
  Creation of new features by creating polynomial combinations of the generated 13 features in step 1 with degree two. Thus, nonlinear features increased the feature dimension from 13 to 105 features. If the expanded features are passed through the classifier directly the accuracy dropped. But with the application of Principal Component Analysis (PCA), the dimension can be reduced to the no of components that is required ---------.
- **Principal Component Analysis (PCA)** [wikipedia]
  The original data is normalized before passing it to the PCA. First few principal components are used to reduce the dimensionality. It is an eigen vector based multivariate analysis. PCA is the orthogonal linear transformation that transforms data into a new coordinate system such that the greatest variance by some scalar projection

lies on the first principal component, followed by second greatest variance lies on the second principal component and so on. It is about choosing the eigen vectors with maximum singular values.

- **Linear Discriminant Analysis (LDA)**
  Both the PCA and the LDA are used for dimensionality reduction but they are not the same. PCA is an unsupervised technique that just find the direction of components that maximizes the variance. But LDA is supervised technique which finds the axes that separates the classes maximum. Thus, in this problem statement LDA performs better than PCA.

CROSS VALIDATION

It is a technique to test the effectiveness of machine learning models.Cross validation keeps a portion of the dataset aside from D_train and keeps it only for validation (Training is not done over Validation set) and the remaining dataset is used for Training the model. Since the data points of a specific user are more correlated, the LeaveOneGroupOut Cross Validation is implemented. Each user consists of 1500 records in the training dataset. This LeaveOneGroupOut Cross Validation would set 1500 records of a specific user aside for validation and the remaining 12000 records of the remaining users are used for training the model. Since the dataset consists of 9 users, the cross validation has 9 folds for each hyperparameter. The best parameter that gives the best solution is selected. The testing data is tested against the model with the best parameter attained by cross validation. Cross Validation done at each fold helps the model to learn from unknown data. It helps in better usage of the data to get improved performance.

CLASSIFIERS

NAÏVE BAYES CLASSIFIER

A naïve bayes classifier is a probabilistic machine learning strategy that is used for classification. The assumption made in Naïve Bayes is that there is conditional independence between every pair of features given the class variable. It is a conditional probability model. Bayes therorm is stated as P(A/B) =P(B/A) P(A)/P(B) where A is the event whose probability is found when the event B is true. The event B is also called as evidence. P(A) is the prior probability i.e the probability of A without knowing the evidence. P(A/B) is the posterior probability of B i.e probability of event A after the evidence B is known.

DATASET USAGE

After feature extraction, the training dataset has 13500 records with 13 dimensions. It is split into train set of 12000 records and validation set of 1500 records (One user records) during the cross-validation technique. The LeaveOneGroupOut cross validation is applied on it to choose the best parameters for the naïve bayes classifier. The hyperparameter that was changed is var_smoothing and cross validation was done for each hyperparameter value.

TRAINING AND CLASSIFICATION

For the implementation of Naïve Bayes Classifier it uses the library GaussianNB from sklearn. naïve_bayes. For the implementation of PCA, PCA from sklearn.decomposition was used. For the implementation of LDA, LinearDiscriminantAnalysis from sklearn.discriminant_analysis is used.

PCA is applied to original 13 features

N_components	5	3	6	10	12	13
Cross Validation Accuracy	0.7002	0.6787	0.7705	0.7903	0.7699	0.7699
Testing Accuracy	0.7457	0.7646	0.7955	0.7283	0.7888	0.7888

PCA is applied after feature expansion by using PolynomialFeatures(13→105)→(105→N_COMPONENTS)

N_components	5	20	30	40	35
Cross Validation Accuracy	0.4657	0.6883	0.7093	0.6743	0.7095
Testing Accuracy	0.5494	0.7855	0.842	0.8439	0.8421

After expansion of features LDA is applied

N_components	5	4	3	6	7	12	20
Training Accuracy	0.9958	0.9958	0.9926	0.9958	0.9958	0.9958	0.9958
Testing Accuracy	0.8427	0.8427	0.7866	0.8427	0.8427	0.8427	0.8427

LDA is applied to the original 13 features space

N_components	4	3	5	6	7	12	20	30
Training Accuracy	0.9427	0.9307	0.9427	0.9427	0.9427	0.9427	0.9427	0.9427
Testing Accuracy	0.8335	0.8264	0.8335	0.8335	0.8335	0.8335	0.8335	0.8335

It is observed that Linear Discriminant Analysis (LDA) is better than Principal Component Analysis (PCA). Both the cross-validation accuracy and the test accuracy are larger in LDA. Applying LDA on extended feature space improves both the cross-validation accuracy and testing accuracy. LDA is a supervised technique that takes class label into consideration. It is a way to reduce the dimensionality and preserve the class discrimination information. LDA first finds the centroid of each class. It tries to maximize the distance between centroid of each class and minimize the variance within a class. In the case of PCA it is unsupervised learning ad it does not require class labels. It generates principal components with the highest variance without considering class information**. The best accuracy was obtained for var_smoothing :0.15199. The best cross validation accuracy is 99.58% and the testing accuracy is 84.27% and the training accuracy is 99.57%.**

It is also observed that after extending the feature space, if the features of 105 dimension are directly passed through the classifier the cross validation accuracy drops to is 88.47% and the testing accuracy is 72.10%. Thus, it is noticed that feature dimension reduction by LDA helped in increasing the accuracy. Feature dimensionality reduction help in selecting the most discriminant features and removing the irrelevant features without loosing much information**. The LDA with n_components =5 has the highest cross validation accuracy and testing accuracy.**

SUPPORT VECTOR MACHINE CLASSIFIER

Support Vector Machine finds the hyperplane in N-dimensional space that distinctly classifies the data points. SVM find the decision boundary to separate the classes by maximizing the margin. Support vector machines are generally binary classifiers. This problem statement is multiclass classification with 5 hand postures. Thus, the SVM used in sklearn inherently using the One Vs All technique to classify the data points. Support Vector Machine chooses the hyperplane that has a maximum margin i.e maximum distance between the data points of different classes. Support vectors are the data points that are closer to the hyperplane and it decides the position and orientation of the hyperplane.

Standardize the features by subtracting the mean and dividing it by standard deviation

Training and Testing accuracy after standardization is given below. The train features are standardized separately using the mean and standard deviation of the train data. The test features are standardized separately using the **mean and standard deviation of the test data.**

{'C': 0.2782559402207124, 'kernel': 'rbf'}
0.9093333333333332
0.8890468742594436

Training and Testing accuracy after standardization. The train features are standardized separately using the mean and standard deviation of the train data. The test features are standardized separately using the **mean and standard deviation of the train data.**

```
{'C': 0.2782559402207124, 'kernel': 'rbf'}
0.9093333333333332
0.9495236741077776
```

We can see a clear increase in the testing accuracy if the way standardization is done is changed. The most appropriate method is to standardize train and test data with mean and standard deviation of the train data.

PREPROCESSING

The only preprocessing technique used was standardization. The model was not performing better for PCA or LDA after expansion of features.

Advantages of Cross Validation

Since the training set is split into a train set and a validation set. The true performance of the model can be estimated by testing it on the validation set. Based on the number of folds, the training set is split into 'N' different combination of the train set and validation set. The mean of the accuracy of all the 'N' folds for each parameter combination is calculated. Thus, the cross validation gives a better accurate estimate of model performance.

DATASET USAGE

After feature extraction, the training dataset has 13500 records with 13 dimensions. It is split into train set of 12000 records and validation set of 1500 records (One user records) during the cross-validation technique. The LeaveOneGroupOut cross validation is applied on it to choose the best parameters for the Support Vector Machine Classifier. Since the number of users is 9, then 9 folds are run on each parameter combination(n). Then the no of fits =9*n. The C values are varied, and the gamma value is set to auto. After the best parameter is obtained, the testing dataset is applied to the model to obtain the testing accuracy. **The best cross validation accuracy was obtained for the parameter C=0.2395 and kernel='rbf'. The testing accuracy is 95.08%. The best cross validation accuracy is 91.13%. The training accuracy is 99.65%.**

Best Result

```
Best parameters
{'C': 0.2395026619987486, 'kernel': 'rbf'}
Best cross validation accuracy
0.9113333333333333
Testing Accuracy
0.9508033556092705
Training Accuracy
0.9965185185185185
```

Observations

Kernel	C	Gamma	Cross Validation Accuracy	Testing Accuracy	Training Accuracy
rbf	10	**0.01**	**0.9429**	**0.86719**	**0.977**
rbf	**1**	**0.01**	**0.9306**	**0.8744**	**0.971**
rbf	**0.3728**	**auto**	**0.9145**	**0.8911**	**0.981**
rbf	**0.2783**	**auto**	**0.9093**	**0.9495**	**0.959**
rbf	**0.3594**	**auto**	**0.9147**	**0.8917**	**0.97**
rbf	**0.2395**	**auto**	**0.9113**	**0.9508**	**0.9965**

The kernel used in SVM classifier is RBF(Radial base Function) also called as the Gaussian Kernel. It was observed that as C value becomes larger the no of misclassification reduced and the cross-validation accuracy increased with increase in C. But after a certain increase in C value the cross validation accuracy decrease or is lesser. The reason being that for smaller values of C a general boundary (soft margin) is obtained which makes the cost of misclassification low. For larger values of C, the boundary produced is more correct by choosing more support vectors to get a (hard margin). The simplicity of the boundary of the SVM reduces with increase in C. But at certain high value of C, the model can start overfitting thus leading to the decrease in testing accuracy with increase of C from the optimal value for C. C is a penalty parameter. Gamma determines the influence of data points on the decision boundary. The higher the gamma the more influence it has on the decision boundary.

KNN CLASSIFIER

K-Nearest Neighbors Algorithm(KNN) is a non-parametric method used for classification. A datapoint is classified to a specific class based on the plurality vote of its neighbors. KNN is based on the algorithm that all similar data points are in proximity. Thus, the data points that are at a closer Euclidean distance are grouped as separate class. In KNN the k nearest neighbors are chosen and a label is assigned based on the closest distance between the points. The sklearn.neighbors.KNeighborsClassifier is used for this project.

Since the total number of users is 9 and we are cross validating based on LeaveOneGroupOut there will be 9 folds and the number of parameters being evaluated is 29 therefore, the total number of fits is 9*29.

PREPROCESSING

The 13 new features are obtained following the Feature Engineering section of the report. Preprocessing techniques like PowerTransformer is used that converts it to a distribution with zero mean and unit variance. The 105 new features are generated using the sklearn's PolynomialFeatures library. Later LDA is applied and the four components are chosen. Thus, the feature dimension is reduced from 105 to 4. **The best cross validation accuracy obtained by this is 99.63%, the testing accuracy is 82.311% and the training accuracy is 99.73%. The best parameters chosen are {'n_neighbors': 19}.**

With preprocessing in the order PowerTranform→PolynomialFeatures→PowerTranform→LDA the testing accuracy drops to 81.65%. Thus, this combination is not used.

n_neighbors	Processing	Cross Validation Accuracy	Testing Accuracy	Training Accuracy
19	PowerTransform→ expansion of features (PolynomialFeatures)→ LDA	0.9964	0.8231	0.9973
14	PowerTransform→ expansion of features (PolynomialFeatures)→ LDA(n_components=4)	0.9950	0.816	0.9962
12	PowerTransform→ expansion of features (PolynomialFeatures)→ LDA(n_components=4)	0.9945	0.79	0.9951

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/#more-on-k

The n_neighbors denotes the number of nearest neighbors in the feature space. This hyperparameter decides the best possible way to fit data. It controls the shape of decision boundary. When the n_neighbors value is small we are restraining the region and not letting the classifier to understand the overall distribution. Thus, the small value of n_neighbors can lead to low bias but high variance. The large values of n_neighbors average the votes thus reducing the effect of outliers. The model produced by the high value of n_neighbors can create smoother boundaries with low variance and high bias.

RANDOM FOREST CLASSIFIER

Random Forest Classifier is a tree-based learning algorithm. It is an ensemble of different decision trees. It aggregates the votes from different decision trees to decide the final class of the testing datapoint.

The performance of the Random Forest Classifier increases with preprocessing using PowerTransform method. The PowerTranform method converts the data into a gaussian distribution by having zero mean and unit variance. The accuracy increased from 72% to 84% with preprocessing.

The hyperparamters like n_estimators, max_features,max_depth and criterion were explored. The cross validation LeaveOneGroupOut was implemented to choose the best parameters that

gives the best accuracy. The function GridSearchCV from sklearn was used to implement cross validation and the best parameters were chosen.

criterion	n_estimators	max_features	Max_depth	Testing Accuracy	Cross Validation Accuracy	Training Accuracy
default	600	Log2	60	0.9064	0.8493	1.0
deafult	600	auto	90	0.8946	0.8498	1.0
default	500	sqrt	40	0.9011	0.8485	1.0
gini	200	Log2	22	0.8918	0.835	1.0
gini	600	sqrt	44	0.8969	0.8486	1.0
default	700	auto	90	0.8989	0.8488	1.0
default	600	Log2	60	0.9064	0.8492	1.0

The n_estimators determines the number of trees in the in the random forest model. Max_depth gives the maximum depth of each tree. With increase in the depth of the random forest, the random forest splits into more nodes and the model will be able to give a more precise accuracy. With larger values of n_estimator also the model would perform well because random forest is an ensemble of decision trees. Thus, a good balance of n_estimators and max_depth can give the best results. **The best testing accuracy was 90.64% and the cross-validation accuracy was 84.92% and the training accuracy was 100%. The best parameters were n_estimators as 600 and max_depth as 60 and max_features as log2.**

LOGISTIC REGRESSION

Logistic Regression classifier is based on the sigmoid function.

The best testing accuracy obtained for Logistic Regression is 88.83% and the training accuracy is 93.8%. Cross validation was not performed for this classifier.

ANALYSIS: COMPARSISON OF RESULTS, INTERPRETATION

Classification report

Naïve Bayes Classifier

	precision	recall	f1-score	support
1	0.90	0.88	0.89	4466
2	0.96	0.92	0.94	4402
3	0.91	0.67	0.77	4779
4	0.65	0.97	0.78	3914
5	0.88	0.78	0.83	3538
accuracy			0.84	21099
macro avg	0.86	0.85	0.84	21099
weighted avg	0.87	0.84	0.84	21099

KNN Classifier

	precision	recall	f1-score	support
1	0.98	0.93	0.96	4466
2	0.89	0.97	0.93	4402
3	0.98	0.71	0.82	4779
4	0.65	0.71	0.68	3914
5	0.64	0.77	0.70	3538
accuracy			0.82	21099
macro avg	0.83	0.82	0.82	21099
weighted avg	0.84	0.82	0.83	21099

SVM CLASSIFIER

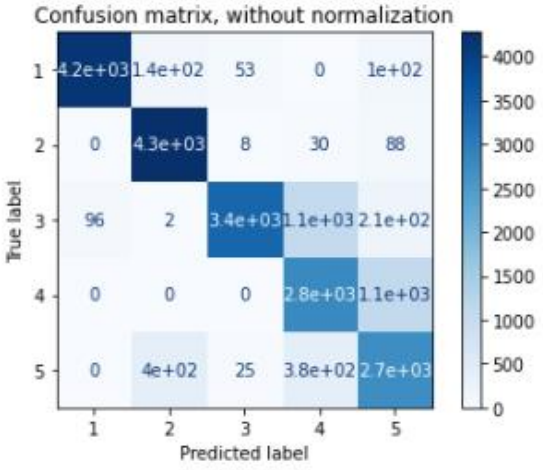
	precision	recall	f1-score	support
1	0.99	0.96	0.98	4466
2	0.98	0.88	0.93	4402
3	0.98	0.98	0.98	4779
4	0.94	0.99	0.96	3914
5	0.86	0.94	0.90	3538
accuracy			0.95	21099
macro avg	0.95	0.95	0.95	21099
weighted avg	0.95	0.95	0.95	21099

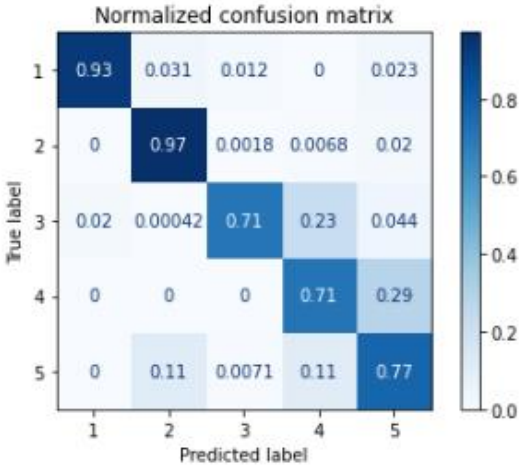
RANDOM FOREST

	precision	recall	f1-score	support
1	1.00	0.97	0.99	4466
2	0.96	0.79	0.87	4402
3	0.97	0.87	0.92	4779
4	0.84	0.94	0.88	3914
5	0.74	0.92	0.82	3538
accuracy			0.90	21099
macro avg	0.90	0.90	0.90	21099
weighted avg	0.91	0.90	0.90	21099

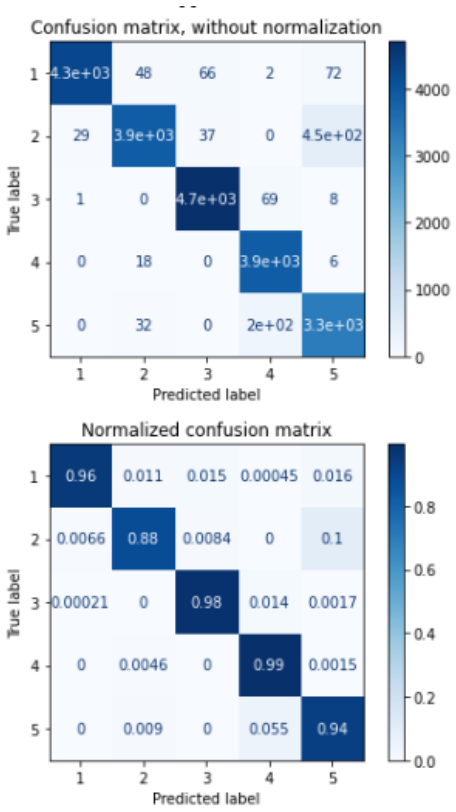
Precision is the measure of (True Positives)/(True Positive+ False Positive). Recall is the measure of (True Positive)/(True Positive+ False Negative). Precision determine the percentage of true positive among the predicted positive. Recall determines the number of actual positive that the model captures. F1 score is a weighted average between precision and recall. F1 score of 1 for best value and score of 0 for worst case. F1 creates a balance between Precision and Recall and is a better measure. Because accuracy can sometimes be affected by False negative and False Positive. Therefore, F1 is a better measure for understanding the performance of the Classifier. The F1 score for SVM is higher than all the other classifiers per class, thus the best classifier.

KNN CLASSIFIER CONFUSION MATRIX

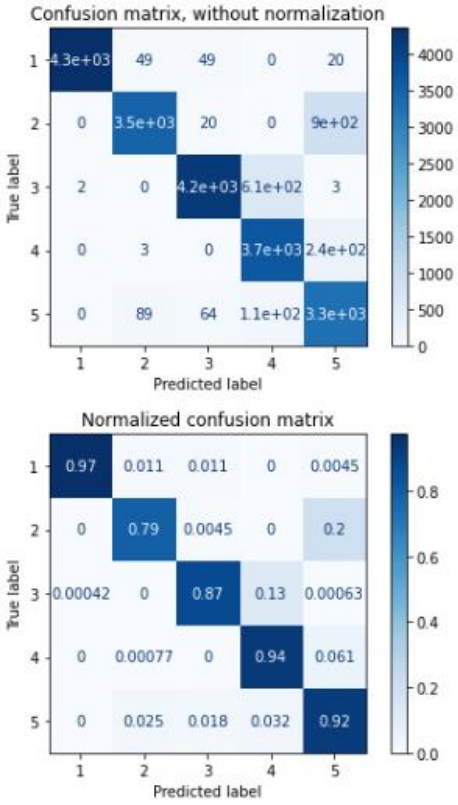




SVM CLASSIFIER



RANDOM FOREST



Confusion Matrix gives the summary of the no of correctly and incorrectly classified datapoints per class. We can also get to know the confusing groups from the confusion matrix by looking at the probabilities.

From the confusion matrix of SVM we can see that the posture 2 has the least probability 0.88. It is more difficult to predict the second posture. The posture 4 has the highest probability 0.99 and the classifier performs the best on posture 4.

From the confusion matrix of Random Forest we can see that the posture 2 has the least probability 0.79. It is more difficult to predict the second posture. The posture 1 has the highest probability 0.97 and the classifier performs the best on posture 1.

From the confusion matrix of KNN we can see that the posture 3 and 4 has the least probability 0.71. It is more difficult to predict the 3 and 4posture. The posture 2 has the highest probability 0.97 and the classifier performs the best on posture 2.

OBSERVATIONS

- We can see that the F1 score for SVM classifier is highest among all. The F1 score of the SVM classifier is 0.98. This shows that the classifier was able to classify all the datapoints to its corresponding class. The precision and the recall values are also large. Thus, there were very few false negatives and very few false positive and overall the classifier performed the best.
- In Support Vector Machine Classifier for smaller values of C a general boundary (soft margin) is obtained which makes the cost of misclassification low. For larger values of C, the boundary produced is more correct by choosing more support vectors to get a (hard margin). Gamma determines the influence of data points on the decision boundary. The higher the gamma the more influence it has on the decision boundary. The best cross validation accuracy was obtained for the parameter C=0.2395 and kernel='rbf'. The testing accuracy is 95.08%. The best cross validation accuracy is 91.13%. The training accuracy is 99.65%.
- In Random Forest Classifier with larger values of n_estimator also the model would perform well because random forest is an ensemble of decision trees. Thus, a good balance of n_estimators and max_depth can give the best results. Thus, optimal number of max_dept and n_estimators gives a better result. The best testing accuracy was 90.64% and the cross-validation accuracy was 84.92% and the training accuracy was 100%. The best parameters were n_estimators as 600 and max_depth as 60 and max_features as log2.
- In KNN Classifier the small value of n_neighbors can lead to low bias but high variance. The large values of n_neighbors average the votes thus reducing the effect of outliers. The model produced by the high value of n_neighbors can create smoother boundaries with low variance and high bias. The best cross validation accuracy obtained by KNN is 99.63%, the testing accuracy is 82.311% and the training accuracy is 99.73%. The best parameters chosen are {'n_neighbors': 19}.
- In Naïve Bayes Classifier the best accuracy was obtained for var_smoothing :0.15199. The best cross validation accuracy is 99.58% and the testing accuracy is 84.27% and the training accuracy is 99.57%. LDA performed better than PCA for dimensionality reduction.

SUMMARY

- The preprocessing method like standardization and Powertransform significantly increased the accuracy of all the classifiers.
- Feature Dimension Reduction Linear Discriminant Analysis performs better than PCA for this problem statement because the components are determined based on classes in LDA.
- The best classifier was Support Vector Machine Classifier with testing accuracy of 95.08%.
- The order of best performing classifiers for this problem statement is SVM Classifier, Random Forest Classifier, Naive Bayes Classifier and K-NN classifier.
- The cross validation of LeaveOneGroupOut is applied to all the classifier and the GridSearchCV of hyperparameters. Thus, for each classifier the best hyperparameters were chosen using cross validation technique. Thus, the results obtained through this gives an almost accurate result. Because, in cross validation the training data is split into training set and the validation set. Thus, the mean of the accuracies obtained in 9 fold (determined by 9 users) for a specific hyperparameter is chosen. And then the hyperparameter with the highest accuracy is chosen as the best parameters. Later the testing data is tested on the model with the best hyperparameters.
- From the classification report we get to know that Support Vector Machine Classifier is the best classifier having high F1 score. The Random forest also has high F1 score. But its performance is not good for all the classes.
- From the confusion matrix we get to know the percentage of datapoints that are correctly classified. We also know that SVM has a consistent good performance for all classes.
- Sometimes Random Forest Classifier can overfit the model. The SVM is performing better on this dataset compared to other classifiers

ACKNOWLEDGEMENTS

- MPR LECTURE NOTES
- Wikipedia
- KNN Classifier https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/#more-on-k
- The libraries for KNN, NAÏVE,SVM,RANDOM forest, GRIDSEARCH for cross validation, confusion plot, classification report are chosen from sklearn website
- Random Forest Classifier [https://towardsdatascience.com/random-forest-classification-and-its-implementation-d5d840dbead0]