DATA MINING
EXERCISE PATTERN PREDICTION
FINAL PROJECT REPORT

We declare that we have completed this assignment completely and entirely on our own, without any consultation with others. We have read the UAB Academic Honor Code and understand that any breach of the Honor Code may result in severe penalties.

We also declare that the following percentage distribution ***faithfully*** represents individual group members' contributions to the completion of the assignment

Name	Overall Contribution (%)	Major work items completed by me	Signature or initials	Date
VENKAT VIVEK YELLANTI	25	EDA and model development	VIVEK	04-22-2021
REVANTH KUMAR NALLURI	25	Model development	REVANTH	04-22-2021
NIKHIL DAMERA	25	Data Preprocessing And grid search CV analysis	NIKHIL	04-22-2021
MANSI SOMAYAJULA	25	Parameter tuning and model Evaluation	MANSI	04-22-2021

ABSTRACT

Exercising has become a habit to many of the individuals and there are many various ways to exercise such as weightlifting, cardio, running etc. People perform exercises in various positions such as sitting-down, standing-up, walking, standing, and sitting. Usually, people will observe systematically as to how much they perform a particular exercise, but they hardly ever quantify how best they do the exercise. The data obtained is not same for every exercise. Here in this project we have opted weight-lifting criterion. Our intent is to apply various data mining classifiers to identify the best classifier for the dataset.

KEYWORDS

Exercise, Classifier, Cross validation, Exploratory Data Analysis, Prediction Models, Confusion Matrix

Introduction

With the rise of life expectancy and ageing of population, the development of new technologies that may enable a more independent and safer life has become a challenge. A positive attitude is a must for successful training. A proper technique influences cardio-respiratory health. Regularly performing muscle building exercises is an important way to improve cardio-respiratory health. And healthy adults should do these exercises because they have been shown to lower blood pressure, increase glucose metabolism, and minimize the risk of cardiovascular disease.

Traditionally, human activity recognition research has focused on distinguishing between various activities or predicting "which" activity was performed at a given time. For the given dataset here, the data is gathered based on how well a person did the activity. Two approaches are commonly used to collect such information and they are: image processing and use of wearable devices. The image processing option requires computational power as well as there are limitations such as user's privacy, and the camera installation and the image quality.

In this project we are trying to build various models with the data collected from wearable devices such as Fitbit, Nike fit band and Jawbone, and find out which model gives us the best result for it.

In Section 2, we will show the EDA process and the necessary steps followed. In Section 3, we will show our results and the level of confidence. In Section 4, we will open it up to a discussion for our findings and explain what we found. Section 5maps out possible directions for future work.

About the Dataset

The data for exercise pattern prediction is collected from wearable devices such as jawbone, Nike fit band and Fitbit. Our aim is to utilize the data which we obtained from accelerometers which were placed in various places such as on fore-arm, belt, arm and dumbbell. We have this information gathered from 6 participants where they were requested to execute barbell lifts both in a correct way and in an incorrect way in 5 different positions as directed such as sitting-down, standing-up, walking, standing, and sitting. The data which we collected is form Kaggle. The dataset has 160 predictors, and the classifiers column is denoted as 'class'. The five postures are denoted as 'A', 'B', 'C', 'D' and 'E'. Each class represents an activity like correct posture while exercising or a wrong posture while exercising. The 160 features are feature engineered by domain specialists.

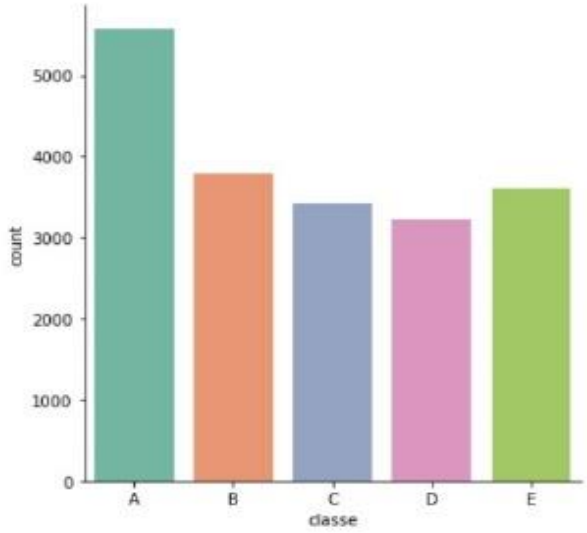


Figure (a) above describes how the gathered data is distributed among each of the five mentioned categories.

Data Cleaning:

Once we compiled our datasets, we had to clean the data by filling in missing values or remove if necessary. The main aim of this step is to identify and handle the incomplete, inaccurate, or irrelevant parts of the data. We eliminated the features which are having null values greater than 20% and used the remaining data for processing.

Categorical variables are usually used to mask the useful information in a data set. It is vital to understand and handle such variables. In our scenario, we have changed 'new_window' and 'user_name' columns to numerical values. This is because, these two features have categorical values while the other features do not contain categorical values.

The data which we used is split into two sets such as training set and testing set. The training set has known output and the model learns this given data to be generalized to other data. While the testing data set is to test the model's prediction in this subset. We performed this activity using Scikit-Learn library with the train_test_split method with a ratio of 70% of training set and 30% of testing set.

Exploratory Data Analysis (EDA):

We used EDA in our project to analyze the data set so that we can obtain their main characteristics/features often using data visualization methods. This is used to determine how best to manipulate data sources to obtain the required answers which makes it easy to discover data patterns, find out anomalies, to test the hypothesis.

EDA is mainly used to understand if the data can reveal beyond the formal modeling or hypothesis testing task and provides better understanding of data variables and the relation between them.

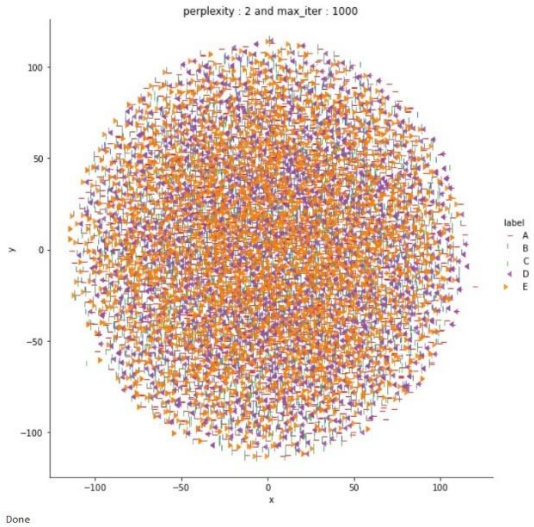
t-SNE stands for t-Distributed Stochastic Neighbor Embedding which is an unsupervised non-linear technique which is mainly used for data exploration and visualizing high-dimensional data.

Here, the aim is to take a set of points in a high-dimensional space and find a faithful representation of those points in a lower-dimensional space, mostly 2D plane.

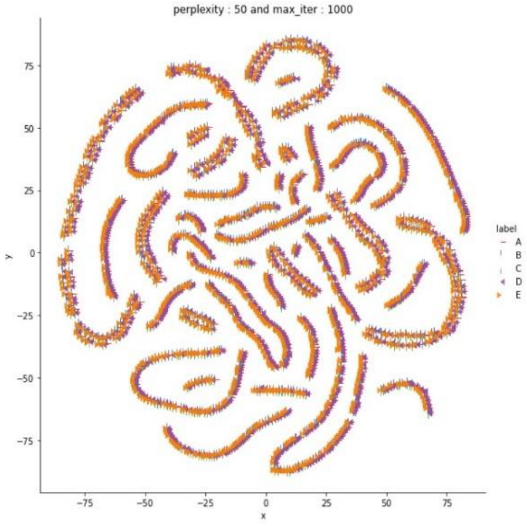
The two tunable hyper-parameters are:

Perplexity: This indicates how to balance attention between local and global aspects of the data. The parameter is a guess about the number of close neighbors each point has.

Here, in our project, we have indicated four perplexity values ranging from 2-50 with maximum iterations of 1000.



Lowest perplexity:2



Highest perplexity : 50

From the above images, even though we are increasing the perplexity, there is no possibility for formation of the clusters. Finally, it expands dense clusters and contracts sparse one, evening out cluster sizes. We can not observe any formation of clusters in the above t-SNE plots.One of the main process in EDA is making pair plots. Pair Plots are easy to visualize relations between each variable. It generates a matrix of relationships between each variable from the data set for an instant examination of the data. It can also be a great jump off point for determining types of regression analysis to use. The features are selected manually for the pair plots. The pair plots are built on histograms and scatter plots. A histogram is a diagonal which allows to see the distribution of a single variable. On the other hand, scatter plots have upper and lower triangles which indicate the relation between two variables. The coloring of the figures in the pair plots helps us to identify the categories easily. We have performed various pair plots. Some of them are mentioned below.

Prediction Models and Results

(i) SVC: The algorithms are harder to visualize as there is complexity in the formulation. SVC also selects the hyperplane which classifies the classes accurately prior to maximizing margin. It is a robust to outliers as it eliminates the outliers while maximizing the margin.

The SVC has a technique known as kernel trick. The SVC kernel is a function which takes the input space and transforms it to high dimension space which converts not separable problem to separable problem.

(ii) KNN: The given k value we can make boundaries of each class. As K value increases the boundary becomes smooth. The error rate at k=1 is always 0.

(iii) Usually, Euclidean distance is taken to measure the distance. KN in reducing overfitting is known. For a given value of K, we perform the square root of the given number of samples in the data set as value of K. It uses grid search cross validation.

(iv) Gaussian Naive Bayes: It is a variant of Naïve Bayes which follows Gaussian normal distribution and supports continuous data. It is simple classification technique but has high functionality. It uses grid search cross validation.

(v) Ada Boost: This is a meta-estimator which begins by fitting classifier on the original dataset which then fits onto additional copies of the classifier on the same dataset and which fits additional copies of the classifier on the same dataset but in scenarios where the weights of incorrectly classified instances are adjusted in such a way that the subsequent classifiers focus more on hard cases. focus more on difficult cases.

(vi) Extremely Randomized Tree: It is also known as Extra Tree algorithm. This works by creating huge number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression.

(vii) Bagging Classifier: It is an ensemble meta- estimator which fits base classifiers each on random subsets of the original dataset and then the aggregate of their individual predictions is taken to form the final prediction. The base estimator to fit on random subsets of the dataset. It uses grid search cross validation.

(viii) Gradient Boosting: It is an iterative functional gradient algorithm which means, it is an algorithms which reduces the loss function by iteratively choosing a function which points towards the negative gradient; a weak hypothesis.

(ix) Decision Tree: It uses multiple algorithms to split a node into two or more sub-nodes. This creation of sub-nodes increases the homogeneity of resultant sub-nodes. The decision tree will split the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

(x) Random Forest: It builds and merges multiple decision trees together to obtain more accurate and stable prediction. Random forest has the same hyperparameters as a decision tree. While the random forest adds additional randomness of the model. It uses grid search cross validation.

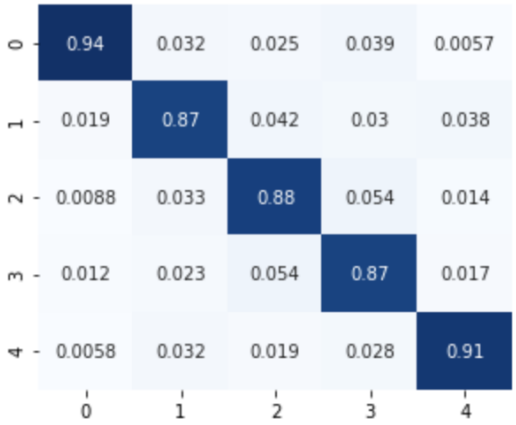
(xi) Random Forest +Bagging Classifier.

(xii) Decision tree + Bagging Classifier

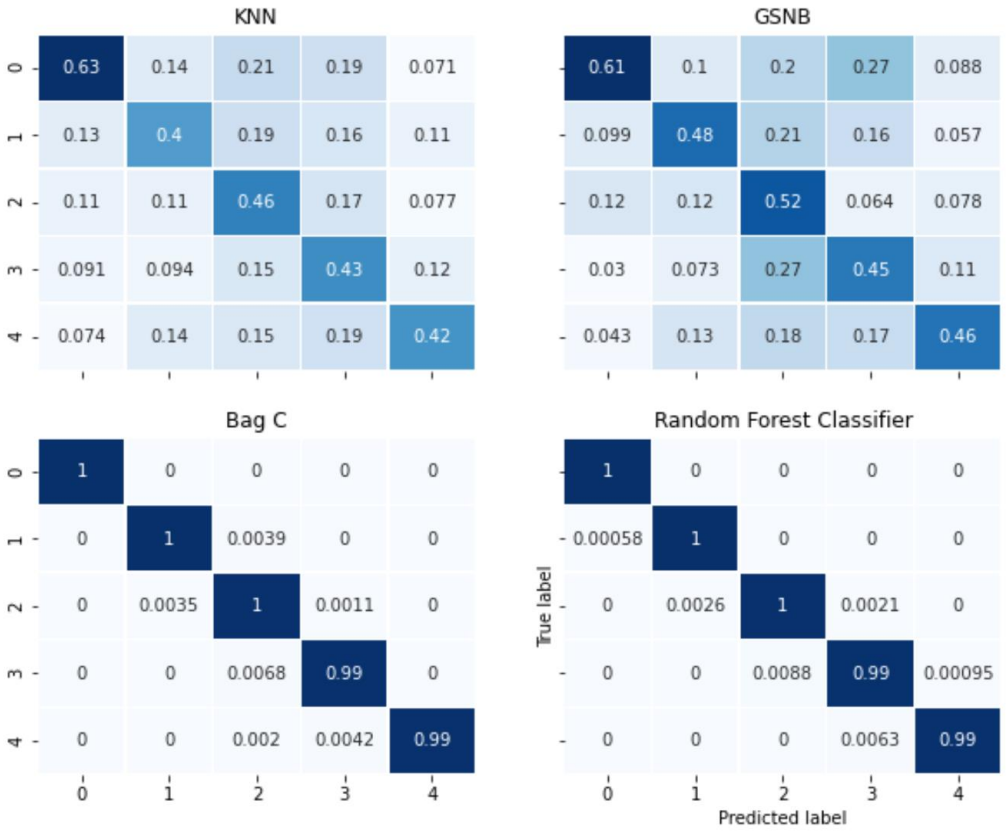
Confusion matrices:


```
Extra Randomised tree Classification

<AxesSubplot:>
```



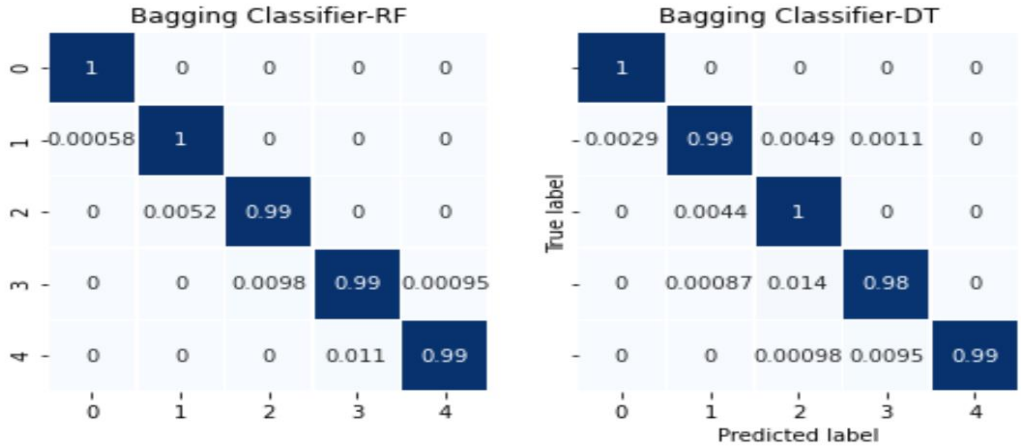
Confusion Matrix for the classifiers before parameter tuning.



Confusion Matrix for the classifiers after parameter tuning

We got best results for bagging classifier, random forest classifier and decision tree. This is the reason why we combined two classifiers form above and developed a new classifier.We combined Bagging classifier and Random Forest Classifier.

Decision Tree and Random Classifier. The following image describes the combined classifiers with best parameter tuning



Confusion Matrix for combined classifiers

Best Parameters after cross Validation:

For KNN:

```
print(clf_knn.best_params_)
```

```
{'leaf_size': 20, 'metric': 'minkowski', 'n_neighbors': 1, 'p': 1, 'weights': 'uniform'}
```

For Gaussian Naïve Bayes:

```
print(Gauss_NB.best_params_)
```

```
{'var_smoothing': 4.328761281083061e-09}
```

For Bagging:

```
print(clf_BaggingC.best_params_)
```

```
{'n_estimators': 59}
```

Randomn Forest:

```
print(clf_RandFC.best_params_)
```

```
{'criterion': 'entropy', 'n_estimators': 19}
```

4. **Discussions and Conclusions**

Based on the results obtained using our dataset, we got the following results. From the above results, it can be concluded that the ensemble methods are best than compared to the other models.

K-nearest neighbors Classifier report

	precision	recall	f1-score	support
A	0.63	0.61	0.62	1773
B	0.40	0.45	0.42	1006
C	0.46	0.39	0.42	1194
D	0.43	0.38	0.40	1083
E	0.42	0.53	0.46	831
accuracy			0.48	5887
macro avg	0.47	0.47	0.47	5887
weighted avg	0.49	0.48	0.48	5887

Accuracy Score 0.48360794971972143

Gaussian Naive Bayes report

	precision	recall	f1-score	support
A	0.61	0.68	0.64	1545
B	0.48	0.53	0.50	1042
C	0.52	0.38	0.44	1412
D	0.45	0.40	0.43	1052
E	0.46	0.58	0.51	836
accuracy			0.52	5887
macro avg	0.50	0.51	0.50	5887
weighted avg	0.52	0.52	0.51	5887

Accuracy Score 0.5163920502802786

Bagging Classifier report

	precision	recall	f1-score	support
A	1.00	1.00	1.00	1711
B	1.00	1.00	1.00	1148
C	1.00	0.99	0.99	1031
D	0.99	0.99	0.99	950
E	0.99	1.00	1.00	1047
accuracy			1.00	5887
macro avg	1.00	1.00	1.00	5887
weighted avg	1.00	1.00	1.00	5887

Accuracy Score 0.9962629522677086

Random Forest Classifier report

	precision	recall	f1-score	support
A	1.00	1.00	1.00	1712
B	1.00	1.00	1.00	1150
C	1.00	0.99	0.99	1027
D	0.99	0.99	0.99	950
E	0.99	1.00	1.00	1048
accuracy			1.00	5887
macro avg	1.00	1.00	1.00	5887
weighted avg	1.00	1.00	1.00	5887

Accuracy Score 0.9962629522677086

Decision Tree Classifier report

	precision	recall	f1-score	support
A	1.00	1.00	1.00	1711
B	0.98	0.99	0.98	1137
C	0.98	0.97	0.98	1033
D	0.98	0.97	0.97	961
E	0.98	0.99	0.99	1045
accuracy			0.98	5887
macro avg	0.98	0.98	0.98	5887
weighted avg	0.98	0.98	0.98	5887

Accuracy Score 0.983692882622728

Random Forest and Bagging Classifier report

	precision	recall	f1-score	support
A	1.00	1.00	1.00	1712
B	1.00	0.99	1.00	1153
C	0.99	0.99	0.99	1027
D	0.99	0.99	0.99	951
E	0.99	1.00	0.99	1044
accuracy			1.00	5887
macro avg	0.99	0.99	0.99	5887
weighted avg	1.00	1.00	1.00	5887

```
print("Accuracy Score",accuracy_score(Y_pred_BaggC_RandF,Y_test))
```

Accuracy Score 0.995243757431629

Decision and Bagging Classifier report

	precision	recall	f1-score	support
A	1.00	1.00	1.00	1716
B	0.99	0.99	0.99	1143
C	1.00	0.98	0.99	1038
D	0.98	0.99	0.99	947
E	0.99	1.00	1.00	1043
accuracy			0.99	5887
macro avg	0.99	0.99	0.99	5887
weighted avg	0.99	0.99	0.99	5887

Accuracy Score 0.9930355019534568

- Results are better when gridsearch is applied because it uses the best poarameters after cross validating. Combining.
- predictions from multiple models in ensembles works better if the predictions from the sub-models are uncorrelated or at best weakly correlated.
- Random forest changes the algorithm for the way that the sub-trees are learned so that the resulting predictions from all of the subtrees have less correlation.

Results obtained for the models.

Classifier	Accuracy	F1 score
KNN	0.30	0.26
KNN-GS	0.48	0.47
SVC	0.29	0.09
GNB	0.51	0.52
GNB-GS	0.52	0.51
ADAB	0.71	0.70
ERTC	0.94	0.93
RF	0.99	0.99
RF + BC	0.99	0.99
DT + BC	0.99	0.99

5. Future Work

By extracting best features from above classifiers, we can develop a new classifier and use it for complicated data.

ACKNOWLEDGMENTS

I would first like to thank my thesis advisor Dr. Chengcui Zhang of the Department of Computer Science at University of Alabama at Birmingham. Dr. Zhang's office was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this paper to be my own work but steered me in the right the direction whenever he thought I needed it.

REFERENCES

[1] https://www.kaggle.com/athniv/exercisepatternpredict

[2] Morbitzer, Christoph and Strachan, Paul and Simpson, Catherine; (2003) Application of data mining techniques for building simulation performance prediction analysis.

Heazlewood, T., & Walsh, J. (2017). Data Mining: Applications of Neural Network Analysis in Exercise and Sport Science. In Proceedings 8th International Conference on Computer Science Education: Innovation and Technology (CSEIT 2017) (pp. 77-83) https://doi.org/10.5176/2251-2195_CSEIT17.42