**Stress Detection using Correlation Clustering based Feature Selection**

ID Name

2015A7PS0076G Kunal Rijhwani

2015A7PS0059G Siddharth Shah

2015A7PS0048G Anshul Aggarwal

2015A7PS0011G Kunal Dewan

**Abstract:**

Feature Selection is useful in many areas requiring intelligent machine learning systems such as anomaly detection, natural language processing, classification and image processing. It is considered highly useful as it can reduce dimensionality of the data without sacrificing the performance of the classifier. Selection of good features is integral to the performance of any classification problem. Thus, feature selection proves highly useful as it improves the computational efficiency of the classifier while still being independent of the classifier. In this report, we present a novel correlation coefficient clustering based feature selection algorithm which retains the most class dependent features in each cluster while removing the other features in the same cluster. The selected features were then used to perform stress detection using data calculated from the wire physical activity tracker (from FitBit). Our model is evaluated using a variety of classifiers like Random Forest Classifier, Support Vector Machine (SVM) and Bernoulli NB Classifier. We have also achieved an improved accuracy over the existing classifiers using ensemble methods. The chief purpose of this project is to use a machine learning approach in stress detection while being as noninvasive as possible for users.

**Approach:**

1. **Preprocessing**

In this project, we preprocessed the data by removing one column in which all values were equal to one same value and then we processed the stress column by converting all the words “high” “medium” and “low” to the same uppercase. There was not much need for further processing as we had mostly numeric data so we carried on ahead with cluster formation on the 26 attributes we had, leaving the “Stress Level” column as that was the value that we had to predict.

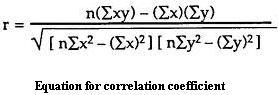
1. **Cluster Formation**

In this project, we use correlation coefficient instead of Euclidean distance as the similarity measure used in the clustering algorithm. This is done to find the most related and mutually unrelated features with respect to the target class. The algorithm used is a partitional K means clustering algorithm which groups the set of features into clusters by computing their correlation coefficient value. Thus, similar features are grouped into the same cluster. To perform dimensionality reduction, we pick the most class dependent feature from each cluster. This is done with the aim of improving the overall classification accuracy.

1. **Feature Extraction**

**3.1) Constructing the initial clusters:**

The correlation between 2 features x and y is calculated using the following formula:



where n is the total number of tuples in the data set. Two variables are said to have a

strong dependence when their correlation coefficient value is close to -1 or +1. A value

of 0 indicates that the variables are totally unrelated. As we are simply looking for a

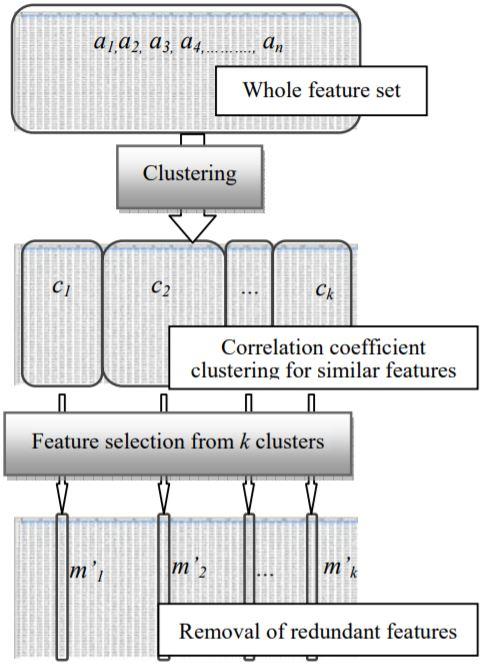
strong dependence (irrespective of negative or positive ) we use the absolute value of r (correlation coefficient ) as a distance measure. Subsequently, clustering analysis divides the entire feature set into different groups.

**3.2) Removing highly correlated terms from a cluster:**

Features in the same cluster are close to each other and we do not need more than 1

features of the same kind to perform classification thus from each cluster we pick only

that feature which has the highest correlation with respect to the target class labels.

*Figure 1. Feature selection process to reduce dimensions.*

**3.3) Generating the training feature vector:**

It is a graph that is often used to identify subsets of input variables that may be most/least relevant to the problem. This can be done by the construction of a random forest of several decision trees. During the construction of these trees, we can calculate how much the error function drops for a variable at each split point. In our classification task, the error function used is the GINI index at each node of the tree.

These drops in error are averaged across all decision trees and the output is used to provide an estimate of the importance of each input variable. The greater the drop when the variable was chosen, the greater the importance. These outputs can help identify subsets of input variables that may be most or least relevant to the problem and suggest possible feature selection sets that one can use along with the feature selection algorithm to form the training vector for the classification task.

**3.4) Generating the training feature vector:**

Using the clustering algorithm, the important terms obtained from each cluster are used obtained to make an array of input features. The top 2-3 features from the variable importance plot are also added to this set. Now the feature set A is the required

reduced training feature vector.

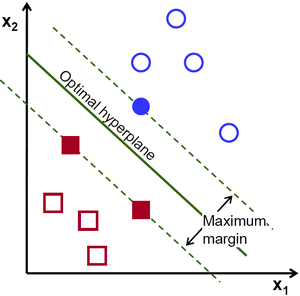
1. **Classification of Data**

**4.1) Support Vector Machine (SVM)**

This classifier is based on support vector based learning. The SVM selects a group of

tuples as its support vectors and uses these to label the classification target. It does so

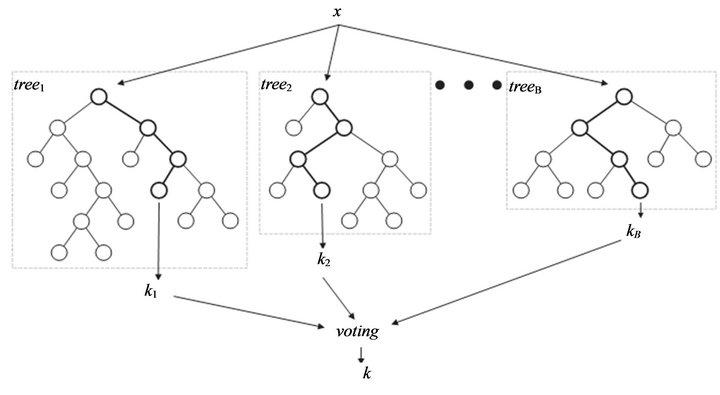
by finding the maximum margin hyperplane that separates records of 2 different classes.



*Figure 2. Illustration of the SVM Classifier.*

**4.2) Random Forest Classifier**

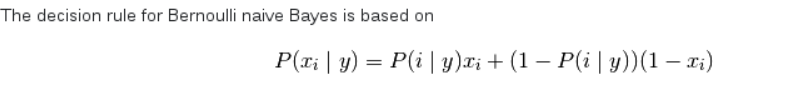
The Random Forest Classifier uses an aggregation of decision tree classifiers to perform the classifications. The input vector is put down in each of the trees of the forest. Each tree gives a classification result and the classifier chooses the classification label having the most votes (among all the labels).

****

*Figure 3. Random Forest Classifier composed of multiple decision trees.*

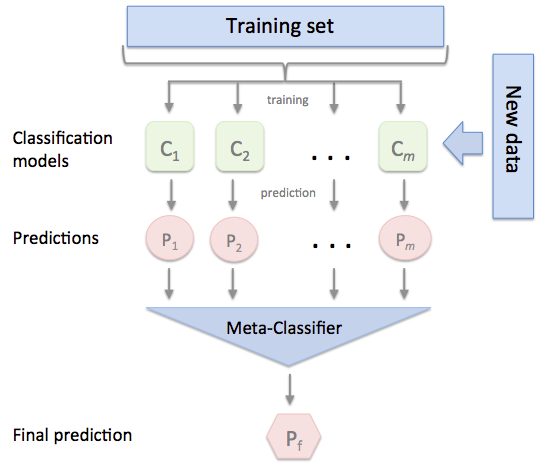
**4.3) Bernoulli NB Classifier**

The Bernoulli model is an alternative to the multinomial model. It is equivalent to binary independence model with 1 indicating presence of a term in a document and 0 indicating absence of the term in the document. The model performs classification by estimating P(t|c) - the fraction of documents of class c that contain term t. The token is then assigned to the class label that yields maximum probability.



## **4.4) Voting Ensemble**

Voting is one of the simplest ways of combining the predictions from multiple machine learning algorithms. It works by first creating two or more standalone models from your training dataset. A Voting Classifier can then be used to wrap your models and average the predictions of the sub-models when asked to make predictions for new data.



*Figure 4. Demonstration of the Ensemble Classifier.*

**Experiment:**

In this experiment, 26 labels were given in the data and labels that can be used as cluster heads to select only the optimal features from the set of features were to be found. We used these to predict the stress levels (the target class). So, pearson r module of python was used to calculate the correlation between the 2 columns of the data frame. There were 26 features excluding the label of “Stress Levels” that were to be predicted. For this, we created a power set of the 26 features and in each iteration, we chose one subset of the 26 features as the head of the clusters and assign the remaining labels to either one of the heads of the clusters.

Then once we make such a cluster, we use the correlation coefficient to check how good the cluster is and then choose the cluster having more intra cluster similarity & lesser inter-cluster similarity. Using this we select clusters with number of heads as 5 being the ones that have the higher correlation coefficient. The variable importance plot is then used to find the labels that are the most important. Following that, the Random forest classifier is used to find the labels that are the most important.

Combining the variable importance plot with the correlation clustering performed, we obtain the following features as the chosen feature sets on which we can further apply our algorithms:

1. Working Hours

2. Minutes sedentary

3. Floors

4. Number of Awakenings

5. BMI.

The pseudo code followed by was as follows:

**Clustering feature selection algorithm**

Randomly select k nodes m(m1,…, mk) from n observations a(a1,…,an);

WHILE originally selected k nodes m(m1,…, mk) != new selected k’ nodes

m’(m1’,…, mk’)

FOR i = 1 to n (observations)

FOR j = 1 to k (nodes)

rj = Correlation\_Coefficient (ai, mj);

IF rj MAX(r1, r2,…, rk) ai belongs to mj’s cluster;

END IF

END FOR

END FOR

FOR p = 1 to k (clusters C1,…,Ck)

FOR q = 1 to Cp’s length t (cluster p’s contents s1,…,st)

rq = Correlation\_Coefficient (sq, Class labels);

IF rq MAX(r1, r2,…, rp)

mp’= sq;

END IF

END FOR

END FOR

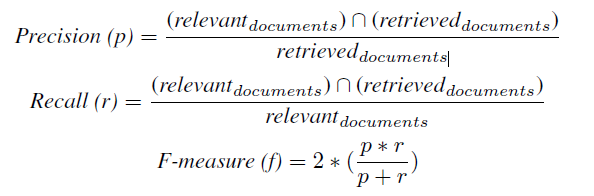
END WHILE

RETURN m’(m1’,…, mk’) ;

Now we use these labels to calculate the confidence after considering only these 5 features and divide our set into training and testing sets.

**Results:**

**Parameters for performance measurement:**



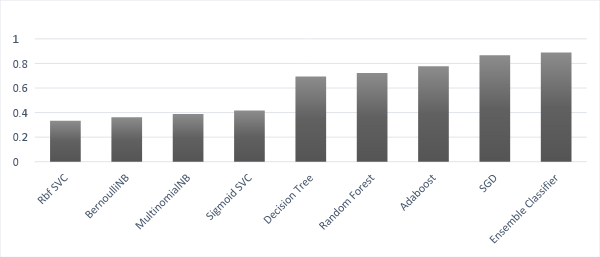
Using the reduced training vector, we now calculated the F-Measure of classification using the standard SVM classifier and achieved an F-Measure of 41.7%. To improve the accuracy, we performed classification using the Bernoulli NB and Random Forest classifier and achieved an F-Measure of 36.11 and 72.2% respectively. To further reduce the variance of our individual classifiers, we used ensemble algorithms such as AdaBoost and Stochastic Gradient Boosting which greatly improved our F-Measure scores to 77.78% and 86.67% respectively. Finally, we also constructed an ensemble classifier consisting of three random forest classifiers, which yielded an F-Measure value of 88.9%.

|  |  |
| --- | --- |
| **Classifier** | **F-Measure** |
| Rbf SVC | 0.333333333 |
| BernoulliNB | 0.361111111 |
| MultinomialNB | 0.388888889 |
| Sigmoid SVC | 0.416666667 |
| Decision Tree | 0.694444444 |
| Random Forest | 0.722222222 |
| Adaboost | 0.777777778 |
| SGD | 0.866666667 |
| Ensemble Classifier | 0.888888889 |

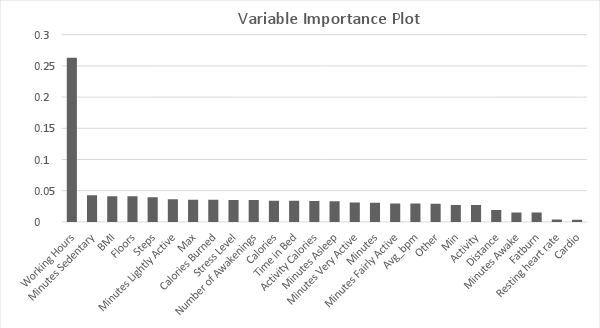
*Table 1. F-Measure obtained from several Classifiers used*

|  |  |
| --- | --- |
| **Labels** | **Importance** |
| Working Hours | 0.263293265 |
| Minutes Sedentary | 0.042574149 |
| BMI | 0.041134674 |
| Floors | 0.041043134 |
| Steps | 0.039424828 |
| Minutes Lightly Active | 0.036466332 |
| max | 0.035416487 |
| Calories Burned | 0.035307676 |
| Stress Level | 0.035196696 |
| Number of Awakenings | 0.035134874 |
| calories | 0.034012443 |
| Time in Bed | 0.033812095 |
| Activity Calories | 0.033651352 |
| Minutes Asleep | 0.033118662 |
| Minutes Very Active | 0.031242584 |
| Minutes | 0.030710531 |
| Minutes Fairly Active | 0.029441951 |
| avg\_bpm | 0.029437392 |
| Other | 0.029282575 |
| Min | 0.027023164 |
| Activity | 0.027008767 |
| Distance | 0.018938218 |
| Minutes Awake | 0.014934399 |
| Fat burn | 0.014909449 |
| resting heart rate | 0.003885035 |
| Cardio | 0.003599268 |

*Table 2. Values computed from the variable importance plot.*



*Figure 5. Plot Showing F-Measure as computed finally*



*Figure 6. Variable Importance plot as computed.*

For sampling purposes, we used the K-cross validation method for sampling without replacement. Out of 448 samples, we trained on 412 samples and tested on remaining 36 samples, to achieve the accuracy as shown in the dataset.

**Innovation**:

As the powerset method is too time consuming , and we need faster computation time, we decided to use a better method. In this method, we followed the same algorithm for k clusters till we selected one feature from each cluster that has the highest correlation with the target label, but then instead of checking the new permutation , we used this result in our next iteration. If the set changes, then we execute the iteration and if it remains the same then we check for k+1 clusters. This computed the same answer very efficiently as well as in a very short amount of time compared to the previous algorithm of checking all the subsets.

**Conclusion**:

Henceforth, based on our feature selection experiment, the important features obtained are Working Hours, Minutes sedentary, Floors, Number of Awakenings and BMI. Now only these features are used in to form the reduced training feature vector for classification to save computation time.

We saw how to use the label importance plot and its effects on the clustering. Using a variety of classifiers on the feature set showed that the Random Forest Classifiers based on decision trees give a better F-Measure than the other learning algorithms. Ensemble Classifiers were also used to further improve the classification as they help in training the minority classes better. Thus, we used a combination of coefficient clustering based feature selection and variable importance plot to for feature selection. This reduced feature set was then used along with ensemble classifiers to achieve a reasonably high F-measure value(88.8%) compared to the standard Support Vector Machine classifiers(44.2%).