

## MASTER OF HEALTH INFORMATICS

**Data Mining & Machine Learning (COMP 809)**

## ASSIGNMENT TWO

## Semester 1, 2018

Submitted by:  
**Farhein Akmal**

**(17977491)**

**Abstract**

Pakinson's disease(PD) is a neurodegenerative disorder of central nervous system that causes partial or full loss of motor reflexes, speech, behaviour, mental processing, and other vital functions. One of the key components of successful treatment is regular monitoring of the patient's status. Unified Parkinson’s Disease Rating Scale (UPDRS) it’s worldwide used scoring system to clinical evaluation and follow up the progression of the disease. The data gathered from the patients can be used to effectively predict the seriousness of the PD’s symptoms. Thus, this assignment uses this dataset and applies 5 models combined with feature-selection, data cleaning and parameter tuning techniques. The goal of this experiment is to achieve the highest possible prediction accuracy for two dependent variable: *motor\_UPDRS* and *total\_UPDRS*. For motor\_uprds, a correlation coefficient of around 0.97 with 17% relative absolute error was expected and for predicting total\_uprds, a correlation coefficient of around 0.95 with 3.4% relative absolute error was expected. Random forest, REPTree, kNN and M5P were performed on the dataset. The goals were achieved with bagging algorithm with parameter turned M5P model.

Table of Contents

[1. Introduction 4](#_Toc517189946)

[2. Background 4](#_Toc517189947)

[2.1 Application Domain 4](#_Toc517189948)

[2.2 Purpose of Study 4](#_Toc517189949)

[3. Experimental Study 5](#_Toc517189950)

[Figure 2. Steps involved in the basic data mining exercise 5](#_Toc517189951)

[3.1 Dataset Description: 5](#_Toc517189952)

[3.1.1 Approach for this dataset 6](#_Toc517189953)

[3.2 Mining Schemes 6](#_Toc517189954)

[3.2.1 Linear Regression 7](#_Toc517189955)

[3.2.2 M5P 7](#_Toc517189956)

[3.2.3 k-Nearest Neighbors 7](#_Toc517189957)

[3.2.4 REPTrees 8](#_Toc517189958)

[3.2.5 Random Forest 8](#_Toc517189959)

[3.3 Pre-Processing 8](#_Toc517189960)

[3.3.1 Preparing Training and Test Dataset 8](#_Toc517189961)

[3.3.2 Outlier Detection 9](#_Toc517189962)

[3.3.3 Missing Values Analysis 9](#_Toc517189963)

[3.4 Feature Selection 9](#_Toc517189964)

[3.4.1 Correlation Feature Selection(Cfs) Subset Evaluator 9](#_Toc517189965)

[3.4.2 Correlation Attribute Evaluator: 9](#_Toc517189966)

[3.4.3 Wrapper Subset Evaluator: 10](#_Toc517189967)

[3.5 Performance Metrics 10](#_Toc517189968)

[3.5.1 Model Evaluation Techniques 11](#_Toc517189969)

[3.5.2 Regression/Prediction Performance Techniques 11](#_Toc517189970)

[3.6 Parameter Turning 12](#_Toc517189971)

[3.7 Boosting Techniques 13](#_Toc517189972)

[4. Results 13](#_Toc517189973)

[4.1 *motor\_UPDRS* – prediction variable 13](#_Toc517189974)

[4.2 *total\_UPDRS* – prediction variable 15](#_Toc517189975)

[5. Conclusion 16](#_Toc517189976)

[References 17](#_Toc517189977)

[Appendix 18](#_Toc517189978)

# 1. Introduction

The objective of this assignment is to have a hands-on experience of data mining and machine learning on real-world dataset. It is to be done by performing experiments using different mining approaches and doing post-experimental analysis. It encourages us to develop and improve our analytical skills in order to achieve the expected goal of the assignment.

This is a report of an attempt to build a model to predict the seriousness of Parkinson disease’s (PD) symptoms with a good accuracy and in less time. This report is divided into three main sections, in the first one I will explain the application domain and the purpose of the study, followed by the typical experimental flowchart which is shown in Figure 1 which is then discussed and concluded.

Figure 1. Ribbon diagram of a typical data-minig exercise

# 2. Background

## 2.1 Application Domain

The mining of information from data though statictics, artificial intellence and database research is called data mining. There are two categories of the application (Gorunescu, 2011):

* Predictive objectives where with the help of given attributes we predict the value of other attributes using techniques like regression, classification, time-series analysis and outlier detection
* Descriptive objectives where it observes the avaliable data’s patterns in order to identifies and categorise data using techniques like sequential pattern discover, association rule and clustering

Parkinson telemonitoring dataset which is extracted from UCI Machine Learning Repository website, is clearly a regression problem because:

* The purpose of this problem is to predict the seriousness of the PD’s symptoms. There are two separate metrics to be predicted:
  + *motor\_UPDRS*
  + *total\_UPDRS*
* These two metrics are continuous and discrete in nature
* It is a numeric prediction problem as all the attributes have numeric values.
* The dataset is multi-varient in nature

## 2.2 Purpose of Study

As mentioned in the assignment guide, for *motor\_UPDRS* variable an accuracy of 97% (or better) can be achieved with 17% of relative absolute error and for *total\_UPDRS*, an accuracy of 95% (or better) can be achieved with around 3.5% of relative absolute error.

Hence, the purpose of this experiment is to achieve the following:

* Find a model(s) and parameter settings to obtains the given accuracy with the lowest relative absoluter error.
* Discover which features can help in obtaining that result.
* Observe how different regression models shows very different results.

# 3. Experimental Study

Several steps were involved in building a good data mining model. A glance of these steps can be seen in Figure 2. This section of the report discusses each stage with a detailed description, figures and tables.

Model 1

Data

Feature Selection

Majority Voting

Model 2

Test Data

Training Data

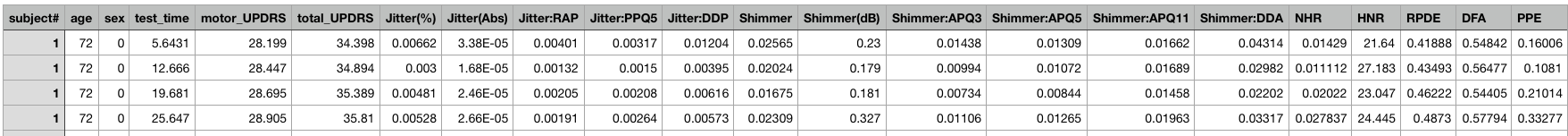
Model 3

Figure 2. Steps involved in the basic data mining exercise

## 3.1 Dataset Description:

The dataset is known as Parkinson Telemonitoring Data Set which is extracted from UCI Machine Learning Repository website. This dataset consists of 5875 instances and 22 attributes. These records includes a range of biomedical measurements from 42 patients (28 men and 14 women) with early-stage Parkinson’s disease(PD) recruited to a six-month trial of telemonitoring device (Tsanas, Little, McSharry, & Ramig, 2010). “The dataset contains 22 attributes including subject number, subject age, subject gender, time interval from baseline recruitment data, motor-UPDRS, total-UPDRS, and 16 biomedical voice measures (vocal features). The vocal features in the dataset are diverse, some of them based on traditional measures (Jitter, shimmer, HNR and NHR), and some (RPDE, DFA, PPE) are based on nonlinear dynamical systems theory. In the dataset, Motor-UPDRS and total-UPDRS were assessed at baseline (onset of trial), three-and-six-month trial periods, but the voice recordings were obtained at weekly intervals. Hence both Motor- UPDRS and total-UPDRS linearly interpolated” (Mohammadi, Hatamlou, & Masdari, 2013).

The downloadable file was converted into CSV format which can be seen in Figure 3. It was then converted into .arff format using WEKA.

  
Figure 3. Sample of Raw Data

Insight of the dataset:

* All 22 attributes are of numeric data type.
* Outliers: There were only 6% of the outliers.
* Missing Values: There were no missing values.
* Multi-variant: The class value for both prediction variables (*motor\_UPDRS* and *total\_UPDRS*) are dependent on more than one attribute. This indicates that the correlations between the attributes is quite high. Thus, doing feature selection can be complex.

### 3.1.1 Approach for this dataset

As there are two variables to predict (*motor\_UPDRS* and *total\_UPDRS*), the Figure 4 briefly outlines the steps planned for this particular dataset. Firstly the available dataset is divided into the train and test set, then after checking and handling noisy data in the train set, feature selection was done for predicting *motor\_UPDRS*. Note, the attribute *total\_UPDRS* needed to be removed when doing feature selection for *motor\_UPDRS*, as the feature selection models will show that the *motor\_UPDRS* is highly dependent on *total\_UPDRS* however the values of *total\_UPDRS* can only be predicted once we know *motor\_UPDRS*. Therefore, once we train our model to predict the variable *motor\_UPDRS* , we then replace the given values to predicted values for *motor\_UPDRS* in the dataset, and then perform pre-processing of dataset again, to predict *total\_UPDRS*.

Figure 4. Flowchart of the steps involved to predict *motor\_UPDRS* and *total\_UPDRS* for Parkinson Telemonitoring Data Set

## 3.2 Mining Schemes

The main factors on which decision of selecting a mining model depends are (Chikohora, 2014):

1. Structure of the dataset
2. Purpose of the data mining
3. Accuracy of the model
4. Time taken to build that model
5. Interpretability efficiency

The observations were done and a plan was made while studying this dataset, are follows:

* The purpose of this problem is to predict how severe are the symptoms. The dependent variables are continuous discrete. Hence, its indicates that it is a regression(prediction) problem.
* The dataset is appropriate for a regression problem. Hence, confirmed.
* Normalisation can be avoided considering the dataset is of regression problem. Also, we will be doing cross validation with 10 folds.
* Feature selection will be done because some of the attributes are more significant in order to do the prediction of the outcome variable i.e. *motor\_UPDRS* and *total\_UPDRS*
* Linear Regression will be applied at first to check the significance and correlations of independent-dependent variable. The accuracy obtain from Linear Regression will be considered as a benchmark for other models, so that we can compare if the result of further applied models are better or not.
* With the purpose of knowing the impactful variables, Random Forest which is an extension of decision tree in an assembled way, will be applied.

Considering the above points, the algorithms that can be considered for this dataset are Linear Regression, M5P, kNN, REPTree and Random Forest.

### 3.2.1 Linear Regression

Linear Regression only supports regression type problems. It works by estimating coefficients for a line that best fits the training data. It is a very simple regression algorithm, which is fast to train and can have great performance if the output variable of the data is a linear combination of the inputs. For Parkinson Telemonitoring Data Set, I applied linear regression before moving onto a complex algorithm while selecting features, because linear regression based feature selection method is promising as it selects discriminative features (Hasan, Hasan, & Mottalib, 2015).

### 3.2.2 M5P

M5 builds a tree for a given instance, to predict numeric values. when the input attributes can be either discrete or continuous the model requires the output attribute to be numeric. For a given instance the tree is traversed from top to bottom until a leaf node is reached. At every node of the tree a decision is formed to follow a specific branch based on a test condition on the attribute related to that node. every leaf node incorporates a linear regression model associated. The tree is called a model tree as the leaf nodes contain a linear regression model to get the predicted output (Polumetla, 2006). we start with a set of training instances to build a model tree, using the M5 model. The tree is built using a divide-and-conquer technique.

### 3.2.3 k-Nearest Neighbors

IBK is a k-nearest-neighbor classifier. It works by storing the entire training dataset and querying it to locate the k most similar training patterns when making a prediction. There is no model other than the raw training dataset and the only computation performed is the querying of the training dataset when a prediction is requested. It is a simple algorithm, but one that does not assume very much about the problem other than that the distance between data instances is meaningful in making predictions.

### 3.2.4 REPTrees

Reduces Error Pruning (REP) Tree Classifier is a fast decision tree learning model and is based on the principle of computing the information gain with entropy and minimizing the error arising from the variance. REPTree is mixture of decision tree and linear regression algorithm, where each leaf node corresponds to a linear regression algorithm. This model constructs the regression/decision tree using variance and information gain. Also, this model prunes the tree using reduced-error pruning with back fitting technique. At the start of the model preparation, it sorts the values of numeric attributes once. (Lakshmi Devasena, 2014)

### 3.2.5 Random Forest

Random Forest is an extension of decision tree in an assembled way. It is a combination of tree predictors where every tree depends on the values of a random vector sampled independently and with the same distribution for all trees within the forest. In standard trees, every node is split using the most effective split among all variables. in a random forest, every node is split using the best among a subset of predictors randomly chosen at that node. This somewhat counterintuitive strategy seems to perform very well compared to several other classifiers and is powerful against overfitting. (Mesarić & Šebalj, 2016). The error rate of a random forest depends on the strength of each tree and correlation between any two trees. It can be used to rank the importance of variables in a regression (or classification) problem in a natural way. It gives better prediction compare to bagged decision trees and there is almost no parameter tuning needed for Random Forest.

## 3.3 Pre-Processing

The real-world data usually includes noises including unwanted or unnecessary attributes, missing data, outliers and extreme values. The regression model can affect on the result which will misguide us and useful information can be easily ignored.

### 3.3.1 Preparing Training and Test Dataset

To build a robust machine learning model, it is important to divide training data and test data before applying any filter or removing any attribute or instances, as then we can play with train data without risking the originality of data, in test data, on which models needs to be checked.

After importing the raw data (in .arff format) in WEKA, remove percentage from unsupervised filters was apllied to split the training and test data, keeping training data 80% (4700 instances) and test data 20% (1175 instances) and setting the invert selection properties to true while applying filter for test data. Important thing to note, at first I took 2/3rd of the data as my training data and 1/3rd as my test data, after seeing the results, I realised 1/3rd of my data for test data is actually a lot of instances because of which my model is not being able to train properly.

### 3.3.2 Outlier Detection

An unsupervised filter determines outlier with no prior knowledge. the data is expected to be statically distributed, pinpoints the most remote points and flag only those data as potential outlier, thus, an unsupervised filter, interquartile range was applied on training set for outlier detection (Hodge & Austin, 2004). There were about 6% of outliers detected which is not much. These outliers were then removed by remove with values filter. After removal of outliers, the total number of instances in the training set were 4383.

### 3.3.3 Missing Values Analysis

This dataset was found to be free of missing columns or fields.

## 3.4 Feature Selection

After pre-processing of the data, we proceed to reduce the dimensionality of our data combining unnecessary variables. A good selection of features can build good predictors that would produce models with high performance (Guyon, Lemaire, Boullé, Dror, & Vogel, 2009). There are two types of feature selection algorithms (Hall, 1999):

1. Filter-based which is known for being fast in computation.
2. Wrapper-based which is known for accuracy gain optimization.

The attribute, “subject#” was manually removed as it clearly acts as a noise in prediction of seriousness of Parkinson’s disease symptoms. For rest attributes, three feature selection algorithms are used:

### 3.4.1 Correlation Feature Selection(Cfs) Subset Evaluator

It is a filter based feature selection algorithm which measures the correlation between the attributes in order to predict an optimal feature set (Hall, 1999). In this experiment CFS was strategized with Best First. Best First starts with no or all features and follows a greedy hill climbing method. According to (Hall, 1999), it increases Decision Tree accuracy quite significantly, which will be extremely suitable for our selection of classifiers. Running it on Parkinson Telemonitoring Data set when predicting

* *motor\_UPDRS* (output variable), selects 11 of the 20 input attributes (note: subject# and *total\_UPDRS* were manually removed)
* *total\_UPDRS* (output varibale), selects 5 of the 21 input attributes (note: *motor\_UPDRS* was replaced by *predicted*\_*motor\_UPDRS*)

### 3.4.2 Correlation Attribute Evaluator:

It is a filter-based feature selection algorithm. that can only be used with a Ranker Search Method, that evaluates each attribute and lists the results in a rank order. It measures Pearson’s correlation between the attribute and the variable in order to assess the value of that attribute (Holmes, Donkin, & Witten, 1994). calculate the correlation between each attribute and the output variable and select only those attributes that have a moderate-to-high positive or negative correlation (close to -1 or 1) and drop those attributes with a low correlation (value close to zero). Running it on Parkinson Telemonitoring Data set for

* *motor\_UPDRS* (output variable) suggest that attributes, “*age*” and “*Shimmer:APQ11*” has the highest two correlation with the output variable
* for total (output variable) suggest that attributes, “*predicted*\_*motor\_UPDRS*” and “*age*” has the highest two correlation with the output variable

A cut-off of 0.170 for relevant attributes for both *motor\_UPDRS* and *total\_UPDRS* is been used.

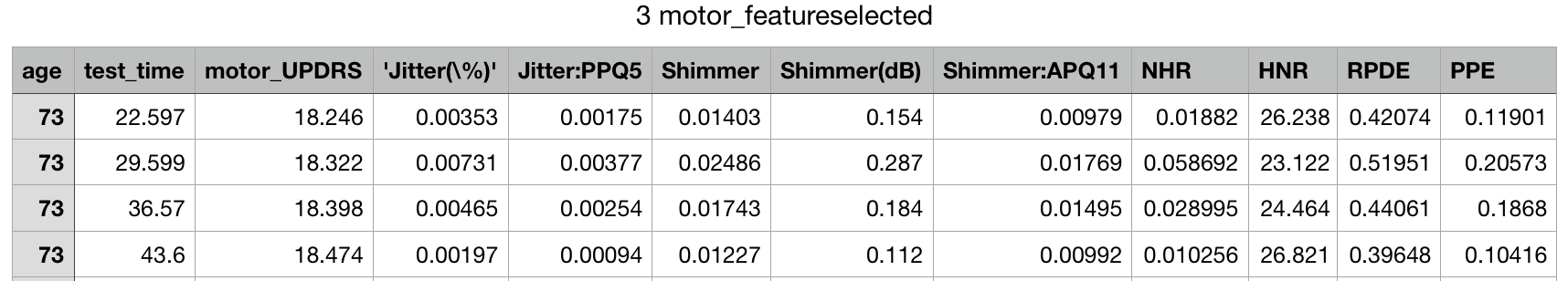
### 3.4.3 Wrapper Subset Evaluator:

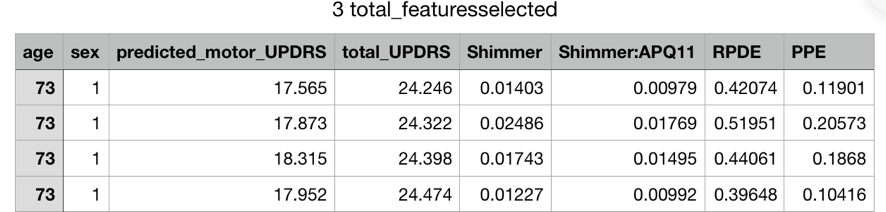
It is a rapper-based feature selection algorithms, which evaluate the performance of the algorithm on the dataset with different subsets of attributes selected. It delivers better accuracy compared to filter methods because feature selection is optimized for a given regression algorithm, in this case its Linear Regression model. (Liu, Motoda, Setiono, & Zhao, 2010). Running this feature selection technique on the Parkinson Telemonitoring Data set when predicting

* *motor\_UPDRS* , selects 16 of the 20 input variables (note: subject# and *total\_UPDRS* were manually removed).
* *total\_UPDRS* (output variable), selects 15 of the 21 input attributes (note: *motor\_UPDRS* was replaced by *predicted*\_*motor\_UPDRS*)

After applying these feature selection models on full training set, results by each models were collected. Features were selected if the attribute was in at least 2 out of these 3 feature algorithm models’ results, when predicting both output variable, *motor\_UPDRS* and *total\_UPDRS*. The screenshot of the result for each feature selection model for both output varibales: *motor\_UPDRS* and *total\_UPDRS* can be seen in appendix A.

The pre-processed training-data after removing the outliers and removing the insignificant features is shown below. The data for predicting *motor\_UPDRS* is shown in Figure 5(a) and for predicting *total\_UPDRS* is shown in Figure 5(b).

  
Figure 5(a). Pre-processed data for predicting *motor\_UPDRS*

  
Figure 5(b). Pre-processed data for predicting *total\_UPDRS*

## 3.5 Performance Metrics

The challenge of predictive modelling is to create models that have good performance in making prediction on new data. Hence it is importance to use strong techniques to train and evaluate your models on the training data and then check on test data. Test options are concerned with estimating the performance of a model on unseen data. In order to create a model that performs best, we must use statistical techniques to best estimate the performance of the model. Such model evaluation and performance estimation techniques are being discussed in this section.

### 3.5.1 Model Evaluation Techniques

Weka offers four model evaluation techniques, that are (Holmes et al., 1994):

* **Training Dataset**: It prepares the model on the entire training dataset, then evaluate the model on the same dataset. It is only to be used when we have all of the data and are interested in creating a descriptive rather than a predictive model. As we have all the data we don’t not need to make new predictions. Our purpose is to create a model to better understand the problem.
* **Supplied Test Set**: It splits the dataset manually using another program. It prepares the model on the entire training dataset and use the separate test set to evaluate the performance of the model. It can be used when the dataset is very large and we don’t need all of it to train a model. Supplied test set is also useful when the test set has been defined by a third party.
* **Percentage Split**: It randomly splits the dataset into a training and a testing partitions each time we evaluate a model. It is good to use to get a quick idea of the performance of a model, but not to be used to make a decision, unless we have very large dataset and knows that the splits sufficiently describe the problem.
* **Cross Validation**: It splits the dataset into k-partitions or folds. It trains a model on all of the partitions except one that is held out as the test set, then repeat this process creating k-different models and give each fold a chance of being held out as the test set. It then calculates the average performance of all k models. This is the gold standard for evaluating model performance, it can be used when we are unsure. It generally provides a more accurate estimate of the performance that the other techniques. Although it is not to be used on a very large data.

While building a model for *motor\_UPDRS* and *total\_UPDRS*, I have used cross validation with fold 10, this will train the model 10 times and hence the overfitting can be avoided. Whereas, when I wanted to predict *motor\_UPDRS* for the full dataset (train + test), to use the predicted *motor\_UPDRS* for next section of the steps (where I predict *total\_UPDRS*), I used training datatset, as at that particular step our purpose was to create a model for better descriptive understanding rather that predictive.

### 3.5.2 Regression/Prediction Performance Techniques

When evaluating a machine learning algorithm on a regression/prediction problem, our results give a number of different performance measures to view like correlation coefficient, mean absolute error, root mean squared error, relative absolute error percent, root relative squared error percent and time taken to build the algorithm. When we evaluate performance summary for regression algorithms, we need to note two output (Frank, Hall, & Witten, 2016):

* Correlation Coefficient: It shows how well the predictions are correlated or change with the actual output value. A value of 0 is the worst and a value of 1 Is a perfectly correlated set of predictions.
* Root Mean Squared Error: It is an average amount of error made on the test set in the units of the output variable. This measure helps to get an idea on the amount of a given prediction may be wrong on average. As obvious, we strive for low error always.

## 3.6 Parameter Turning

For any given algorithm, on the dataset and the purpose of mining, tuning certain parameters will always generate better results. In order to fine-tune these parameters, it is necessary to understand role of each parameter affecting the outcome. Some of the parameter optimization applied to above discussed algorithms are discussed below.

M5P’s parameters include the minimum number of instances per leaf whether to use unpruned tree, and whether to build regress tree rather than a model tree.

IBk is a k-nearest neighbor classifier. Its parameters include the number of newest neighbors, the window size or the maximum number of training instances maintained, whether to weight neighbors by 1-their distance, and other to minimize mean squared error rather than mean absolute error for numeric predictions.

REPTree’s parameters include the minimum number of instances per leaf, the number of folds for reduced error pruning, the seed for random data shuffling and whether to use unpruned tree.

As mentioned previously, there is almost no need for parameter tuning in random forest, hence in this experiment, the random forest was built with default parameters. For M5P, IBK and REPTree several parameters were tuned in this experiment to understand the effect of each parameter on the result. Table 1 shows the default parameters for the above-mentioned models and the tuned parameters for the same.

  
Table 1. Selected Mining Algorithms

## 3.7 Boosting Techniques

In order to make reliable decisions, numerous learningtechniques are combined to form an ensemble of models. Most famous ensemble schemes are bagging, boosting and stacking. The generalization ability of an ensemble model is more reliable than individual algorithms.   
Bagging creates bootstrap samples of training set using sampling, changing the distribution of training set. Boosting changes the distribution of training set adaptively primarily based on the classifiers built earlier (Kotsiantis & Pintelas, 2004). Boosting works efficiently on datasets with strong variance. However, ensemble will lead to lose of interpretability as there are several unique samples on which the classification is applied.   
Stacking is also technique to combine a couple of learning algorithms. Unlike bagging and boosting, stacking is used to combines models of different types. It estimates expected error of each algorithm by cross validation and chooses the best algorithm among the stack of algorithms for build a model (Witten, Frank, Hall, & Pal, 2016). Stacking uses “meta learner” to learn the efficient way of combining the output of base learner algorithms.

For this assignment, we were required to use meta-learner on the best algorithm with best parameter tuning. Hence, I have performed bagging, stacking and boosting with the best model for both output variables.

# 4. Results

This section will discuss the results obtained after conducting the experiment.

## 4.1 *motor\_UPDRS* – prediction variable

Various models were built on 10 folds cross validation to predict *motor\_UPDRS* on the pre-processed train data. The results were noted down (Table 2). Observation highlighted in white were seen to be quite close to the expected value which was 0.97 correlation coefficient with 17% of relative absolute error.

This result of 0.949 correlation coefficient with 16.49% was obtained by M5P model with unsmoothed prediction true. The correlation in model REPTree and Random Forest were close to the targeted correlation, but the relative absolute error value for Random Forest was quite high. Hence they were not considered in the later steps.

A number of parameter turning were done to observe the how each parameter effects the result. Some insights from these observation are:

* In M5P, the correlation coefficient remains roughly similar with parameter turning but as the unsmoothed prediction is changed to true, the relative absolute error is dropped by 5%. The highest error was seen when M5P were built on regression tree.
* In kNN, when the maximum number of training instanced maintained was changed the result went drastically down.
* In REPTree, the lowest relative absolute error was obtained when the minimum number of instances per leaf was 1.

  
Table 2. Results of models build with parameter turning for *motor\_UPDRS*

As seen, M5P with unsmoothed prediction true, show the best results compare to other models. Hence it was combined with bagging, stacking and boosting. Observations for the same is shown in Table 3. The targeted results were achieved when bagging was performed. Bagging creates separate samples of the training dataset and creates a classifier for each sample. The results of these multiple classifiers are then combined (such as averaged or majority voting). The trick is that each sample of the training dataset is different, giving each classifier that is trained, a subtly different focus and perspective on the problem.

  
Table 3. Results of the meta-learner models for *motor\_UPDRS*

After deciding on the final (best) model, which was Bagging+M5P with unsmoothed prediction true, the same model was built on the test data. The results obtained were quite better that what was expected. Hence, the expectation for predicting *motor\_UPDRS* was achieved. The result is shown in Table 4 and the result is attached in appendix B.

  
Table 4. Results of the best model on test data for *motor\_UPDRS*

To predict *total\_UPDRS*, the predicted values of *motor\_UPDRS* had to replace the given value for *motor\_UPDRS*. Hence the best model was then applied on the full original dataset (after manually removing *total\_UPDRS*). The result is shown in Table 5 and the screenshot of the output of predicted\_motor\_UPDRS is attached in appendix C.

  
Table 5. Results of the best model on full data for *motor\_UPDRS*

## 4.2 *total\_UPDRS* – prediction variable

Various models were built on 10 folds cross validation to predict *total\_UPDRS* on the new pre-processed train data which includes predicted *motor\_UPDRS*. The results were noted down (Table 6). Observation highlighted in white were seen to be quite close to the expected value which was 0.95 correlation coefficient. The expected relative absolute error was 3.5% but none of the model built was able to achieve any close value. The correlation coefficient obtained by all the mentioned models were either achieved or were very close, but the relative absolute error value was quite high for all the models. The best relative absolute error was 11.8%, hence M5P with unsmoothed prediction true was considered the best model amongst the others.

  
Table 6. Results of models build with parameter turning for *total\_UPDRS*

M5P with unsmoothed prediction true was then combined with bagging, stacking and boosting, the observation is shown in Table 7. The little increase in the correlation coefficient and decrease in the relative absolute error was seen after bagging. Hence, bagging with M5P with unsmoothed prediction true, was decided on the final model to predict *total\_UPDRS*

  
Table 7. Results of the meta-learner models for *total\_UPDRS*

The final (best) model was tested on the test data. The results obtained where satisfactory as the correlation coefficient of 0.99 were achieved with 3.7 relative absolute error, which were better that the targeted result. The observation is shown in Table 8 and the result is attached in appendix D.

  
Table 8. Results of the best model on test data for *total\_UPDRS*

Finally the best model was then applied on the full original dataset (with predicted *motor\_UPDRS*). The result is shown in Table 9 and the screenshot of the result of predicted *total\_UPDRS* is attached in appendix E.

  
Table 9. Results of the best model on full data for *total\_UPDRS*

# 5. Conclusion

The purpose of this research was to predict the seriousness of the Parkinson Disease’s symptoms. This datamining provided alternative ways to achieve the targeted accuracy and relative absolute error for the two output varibales, *motor\_UPDRS* and *total\_UPDRS,* in the Parkinson telemonitoring dataset. The dataset was multivariant and had no missing values, although there were a few of the outliers which were removed. The one of the variable which were to be predicted, *total\_UPDRS* were dependent on the other output/ prediction variable ie. *motor\_UPDRS*, but it was not vise versa. The result obtained by M5P model with parameter turning were better that REPTress, Random Forest, kNN and Linear Regression. Finally by using boosting technique on M5P, the performance of algorithm was greatly improved and the targeted result were achieved for both output variable: *motor\_UPDRS* and *total\_UPDRS*, with 0.99 correlation coefficient for both and relative absolute error of 2.3% while predicting *motor\_UPDRS* and 3.7% while predicting *total\_UPDRS.*

# References

Chikohora, T. T. (2014). A study of the factors considered when choosing an appropriate data mining algorithm. *International Journal of Soft Computing and Engineering (IJSCE), 4*, 42-45.

Frank, E., Hall, M., & Witten, I. (2016). The WEKA workbench. *Data mining: Practical machine learning tools and techniques. Burlington: Morgan Kaufmann*.

Gorunescu, F. (2011). *Data Mining: Concepts, models and techniques* (Vol. 12): Springer Science & Business Media.

Guyon, I., Lemaire, V., Boullé, M., Dror, G., & Vogel, D. (2009). Analysis of the kdd cup 2009: Fast scoring on a large orange customer database*JMLR. org.* Symposium conducted at the meeting of the Proceedings of the 2009 International Conference on KDD-Cup 2009-Volume 7

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Hasan, M. A., Hasan, M. K., & Mottalib, M. A. (2015). Linear regression–based feature selection for microarray data classification. *International journal of data mining and bioinformatics, 11*(2), 167-179.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review, 22*(2), 85-126.

Holmes, G., Donkin, A., & Witten, I. H. (1994). Weka: A machine learning workbench*IEEE.* Symposium conducted at the meeting of the Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on

Kotsiantis, S., & Pintelas, P. (2004). Combining bagging and boosting. *International Journal of Computational Intelligence, 1*(4), 324-333.

Lakshmi Devasena, C. (2014). Comparative Analysis of Random Forest, REP Tree and J48 Classifiers for Credit Risk Prediction Symposium conducted at the meeting of the International Journal of Computer Applications (0975-8887), International Conference on Communication, Computing and Information Technology (ICCCMIT-2014)

Liu, H., Motoda, H., Setiono, R., & Zhao, Z. (2010). Feature selection: An ever evolving frontier in data mining Symposium conducted at the meeting of the Feature Selection in Data Mining

Mesarić, J., & Šebalj, D. (2016). Decision trees for predicting the academic success of students. *Croatian Operational Research Review, 7*(2), 367-388.

Mohammadi, P., Hatamlou, A., & Masdari, M. (2013). A comparative study on remote tracking of Parkinsons disease progression using data mining methods. *arXiv preprint arXiv:1312.2140*.

Polumetla, A. (2006). *Machine learning methods for the detection of RWIS sensor malfunctions*. Citeseer.

Tsanas, A., Little, M. A., McSharry, P. E., & Ramig, L. O. (2010). Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. *IEEE transactions on Biomedical Engineering, 57*(4), 884-893.

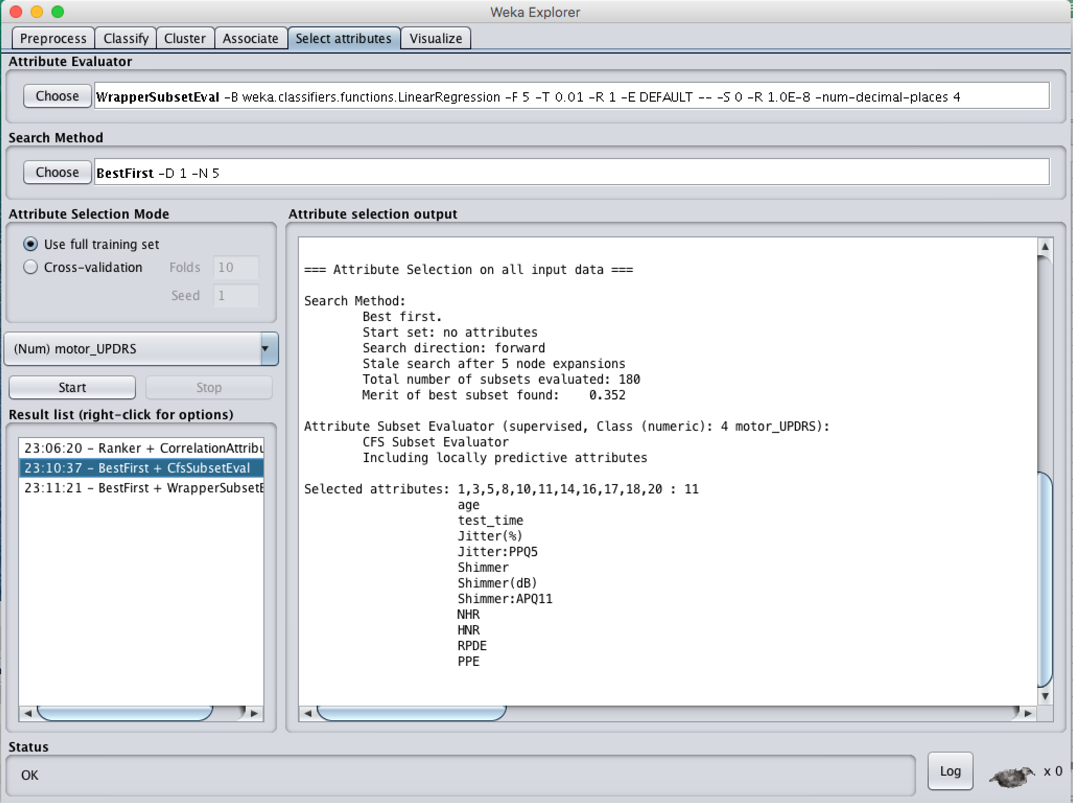
Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann.

# Appendix

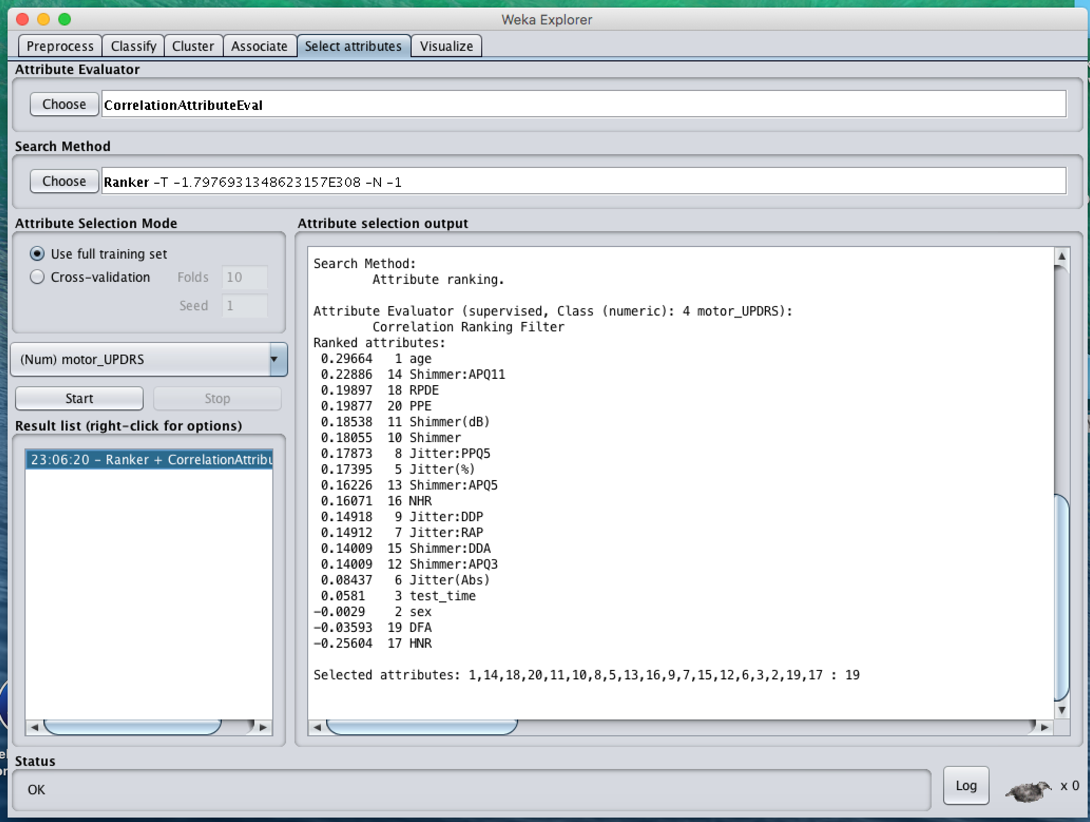
**A. Feature Selection**

A.1 Feature Selection for ***motor\_UPDRS***

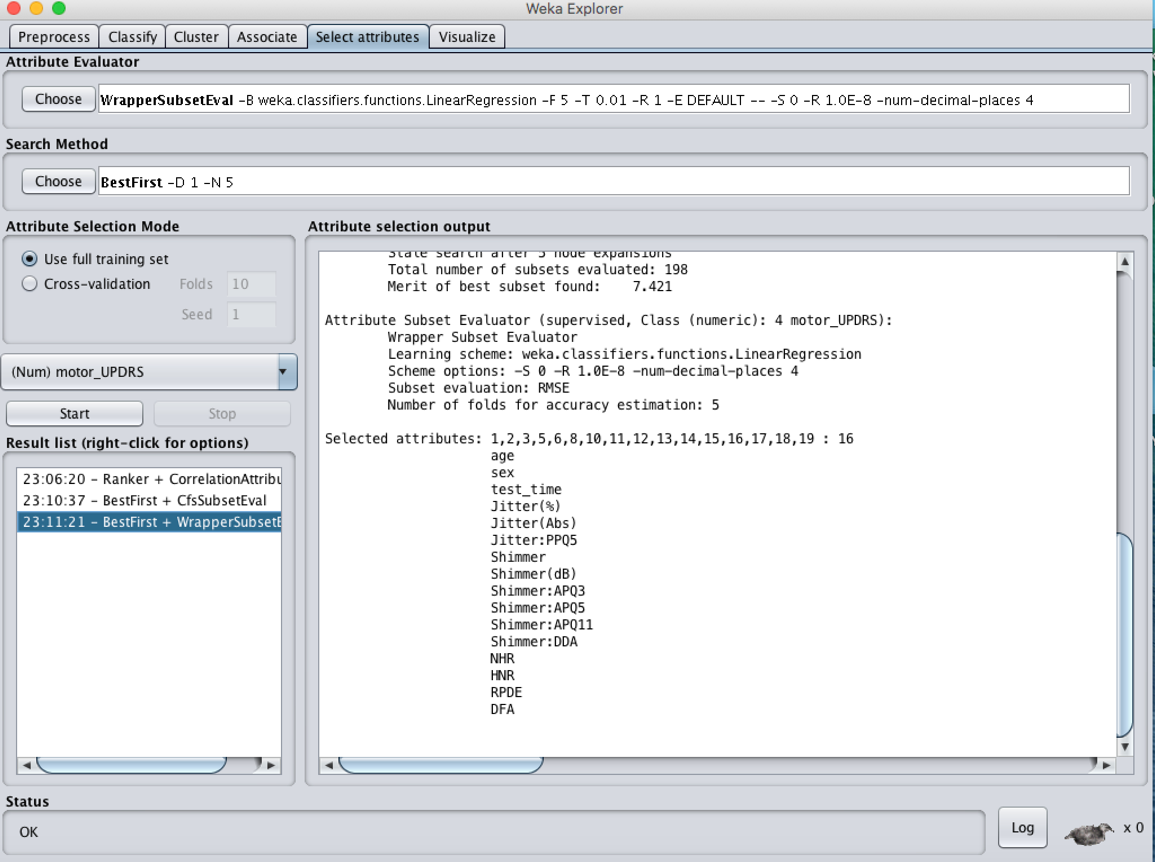
**CfsSubsetEval + BestFirst**



**CorrelationAttributeEval + Ranker**

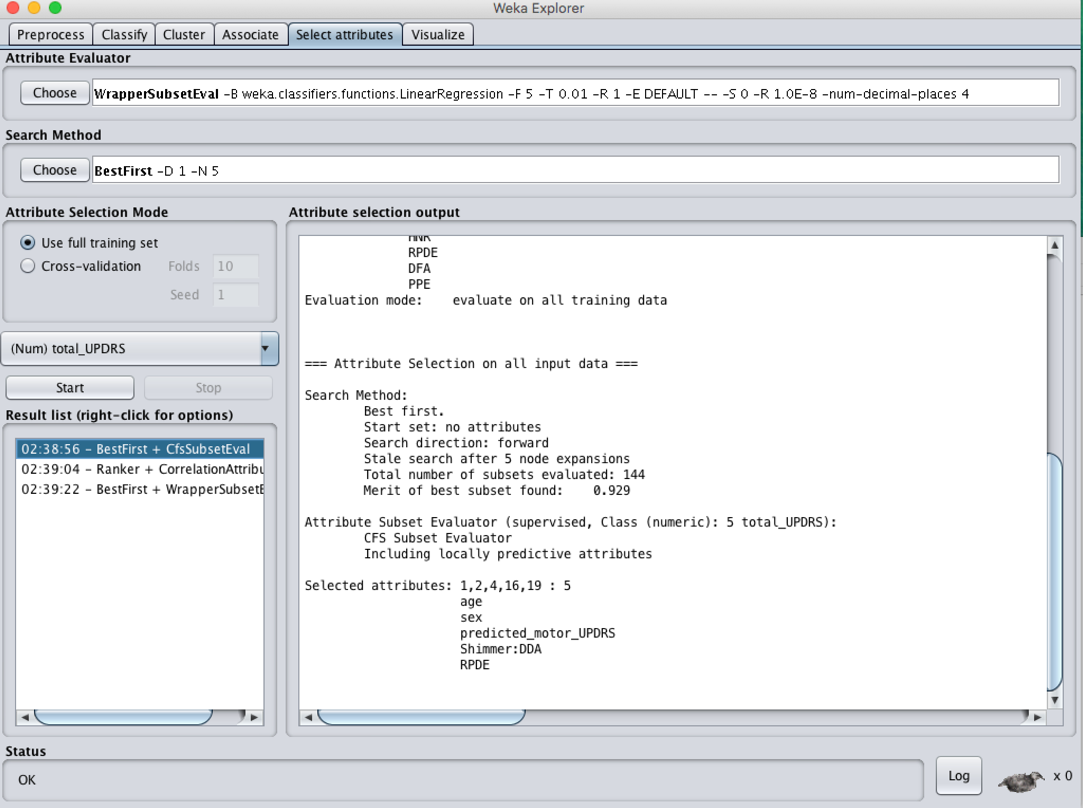


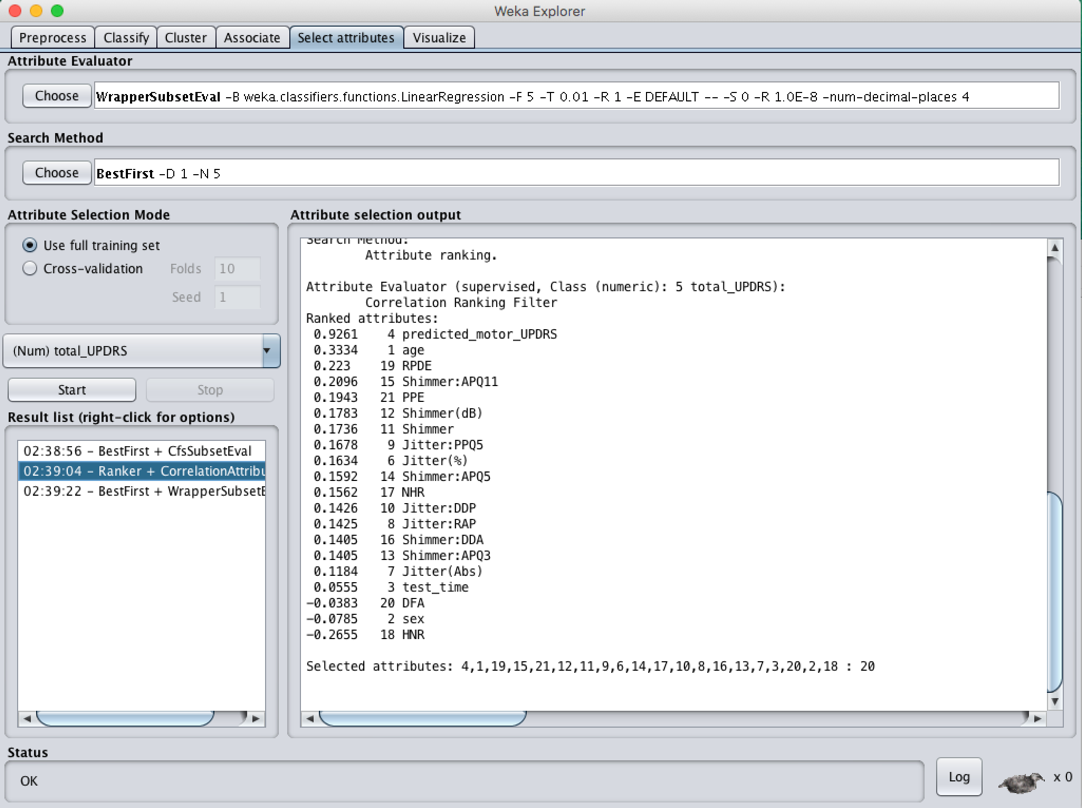
**WrapperSubsetEval + BF + Linear Regression**



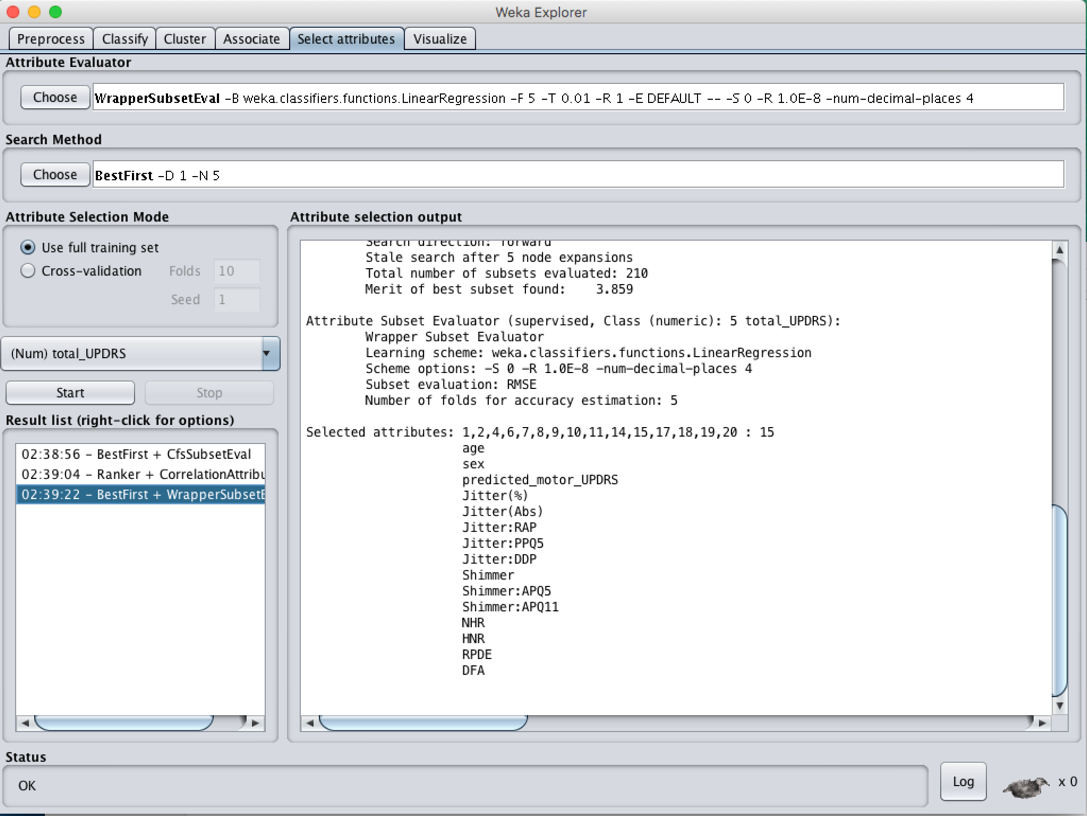
A.2 Feature Selection for ***total\_UPDRS***

**CfsSubsetEval + BestFirst**

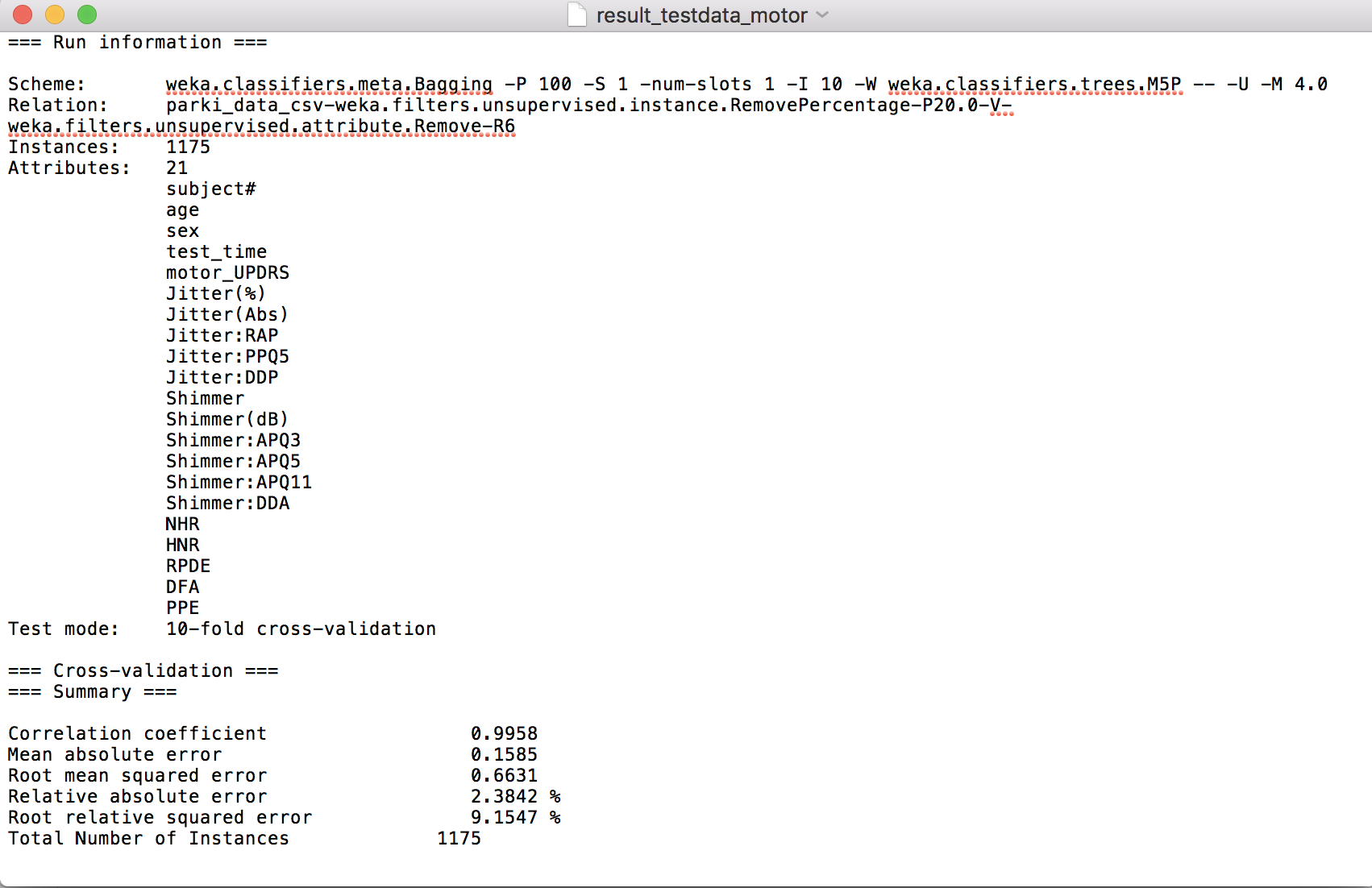


**CorrelationAttributeEval + Ranker**

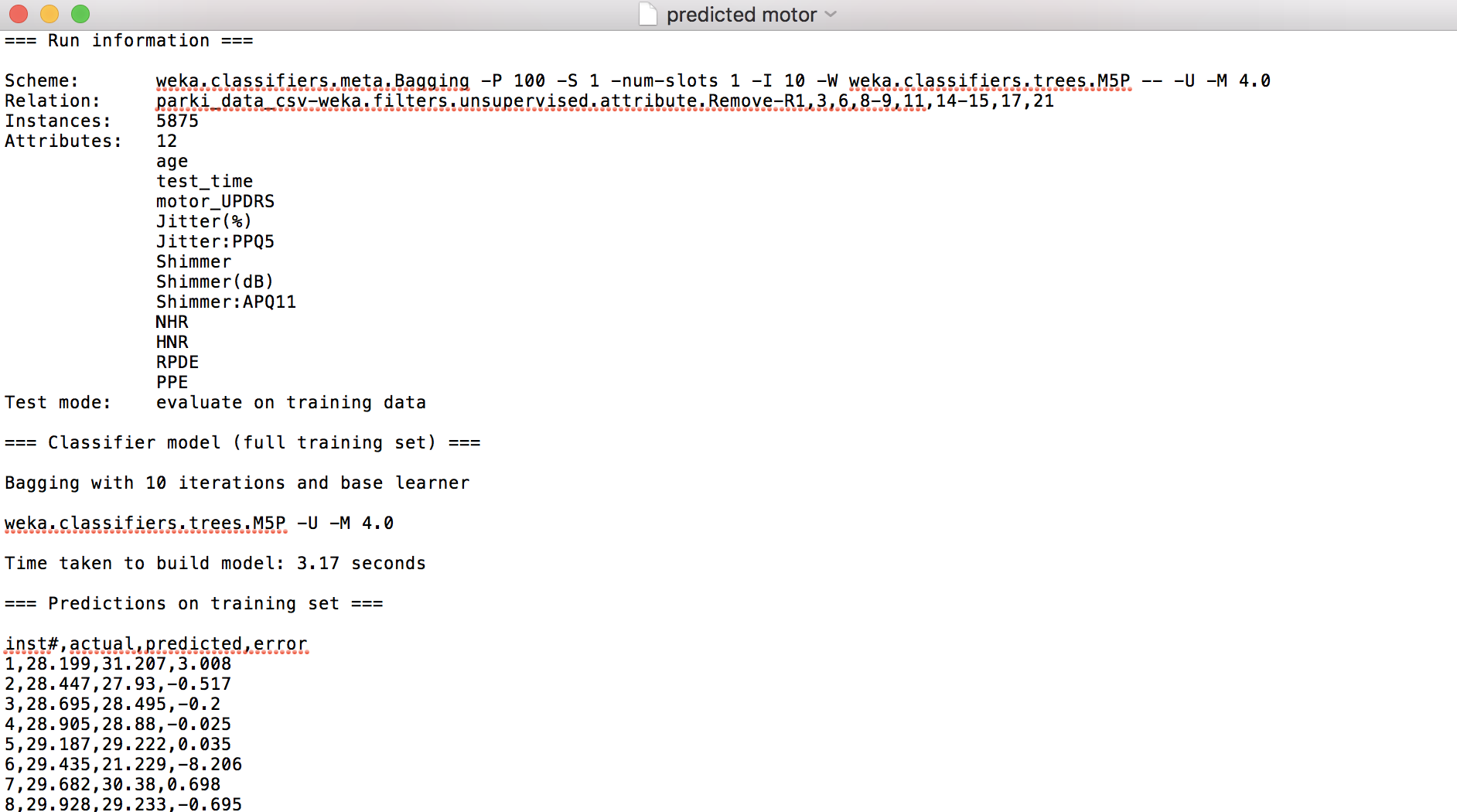
**WrapperSubsetEval + BF + Linear Regression**



**B. Bagging+M5P on test data for *motor\_UPDRS***

****

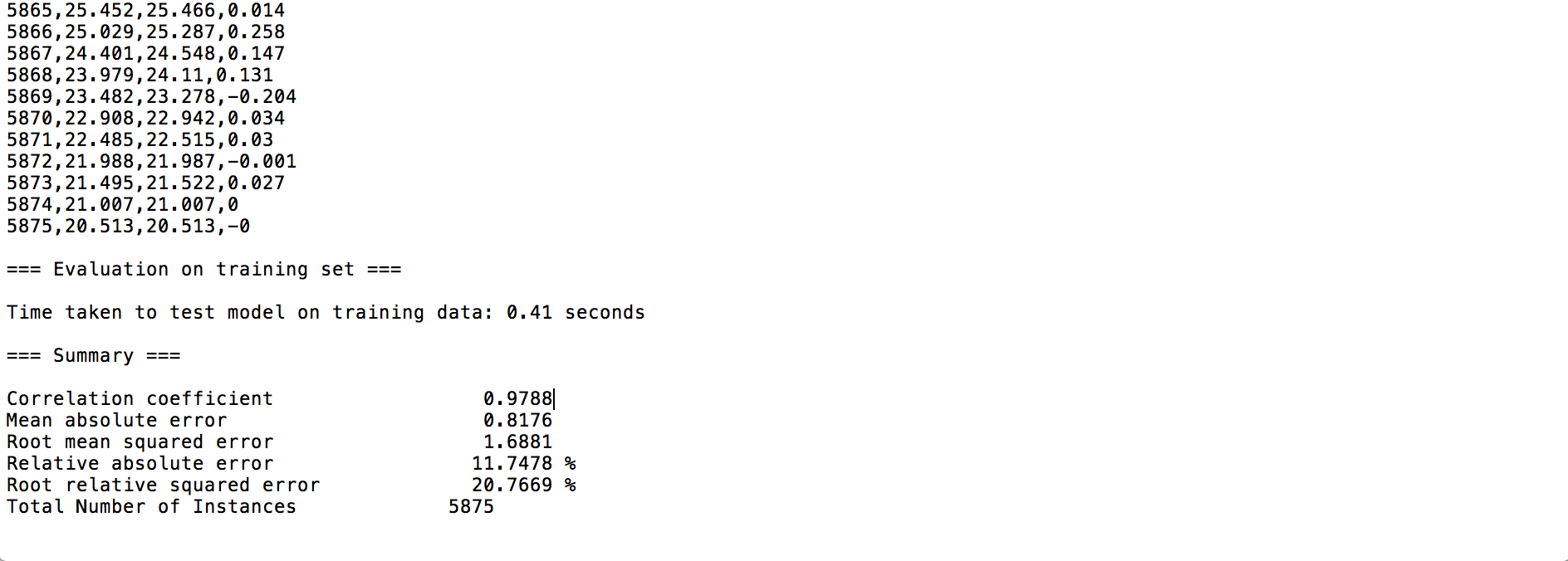
**C. Predicted *motor\_UPDRS* result**



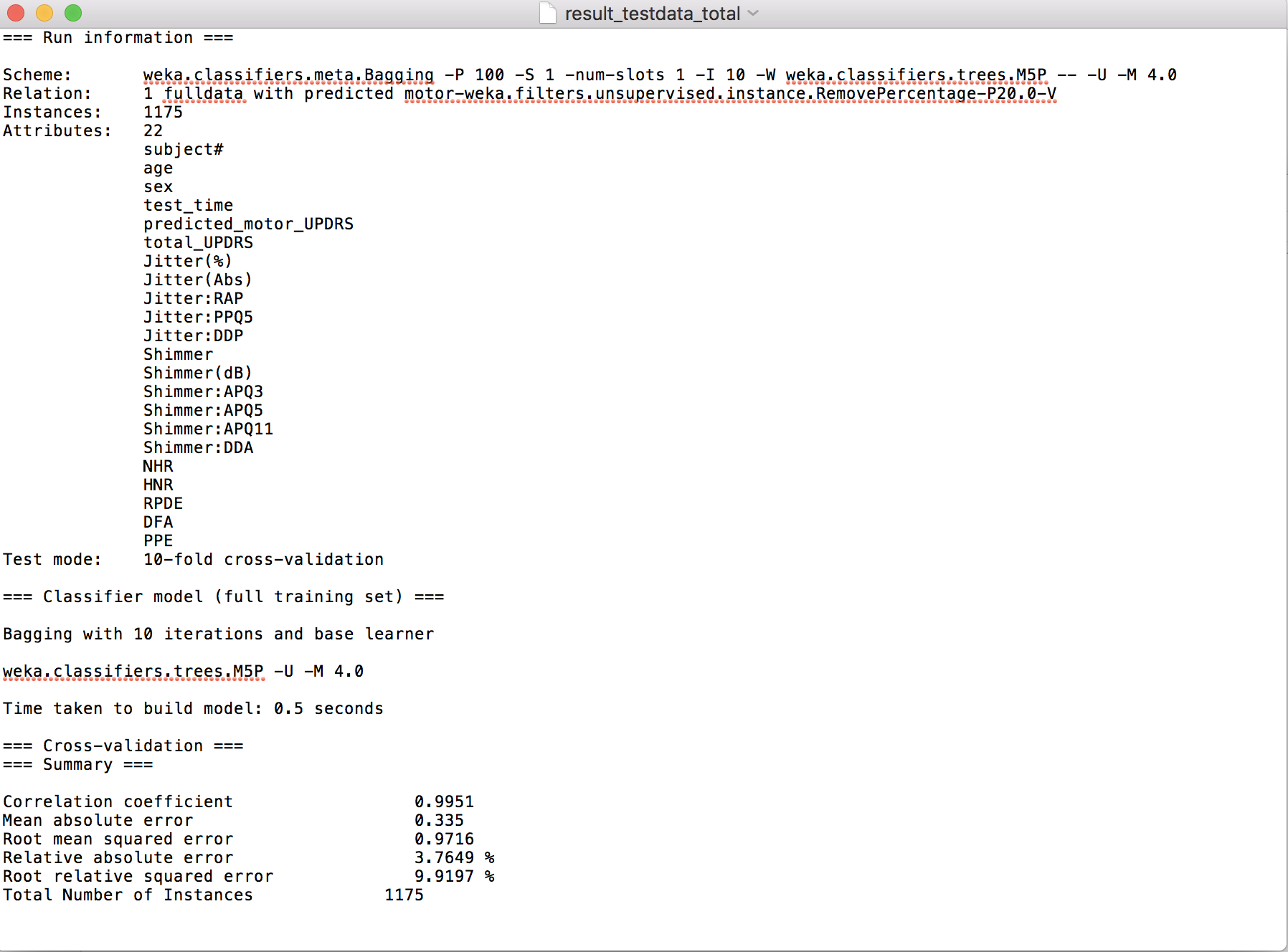
.

.

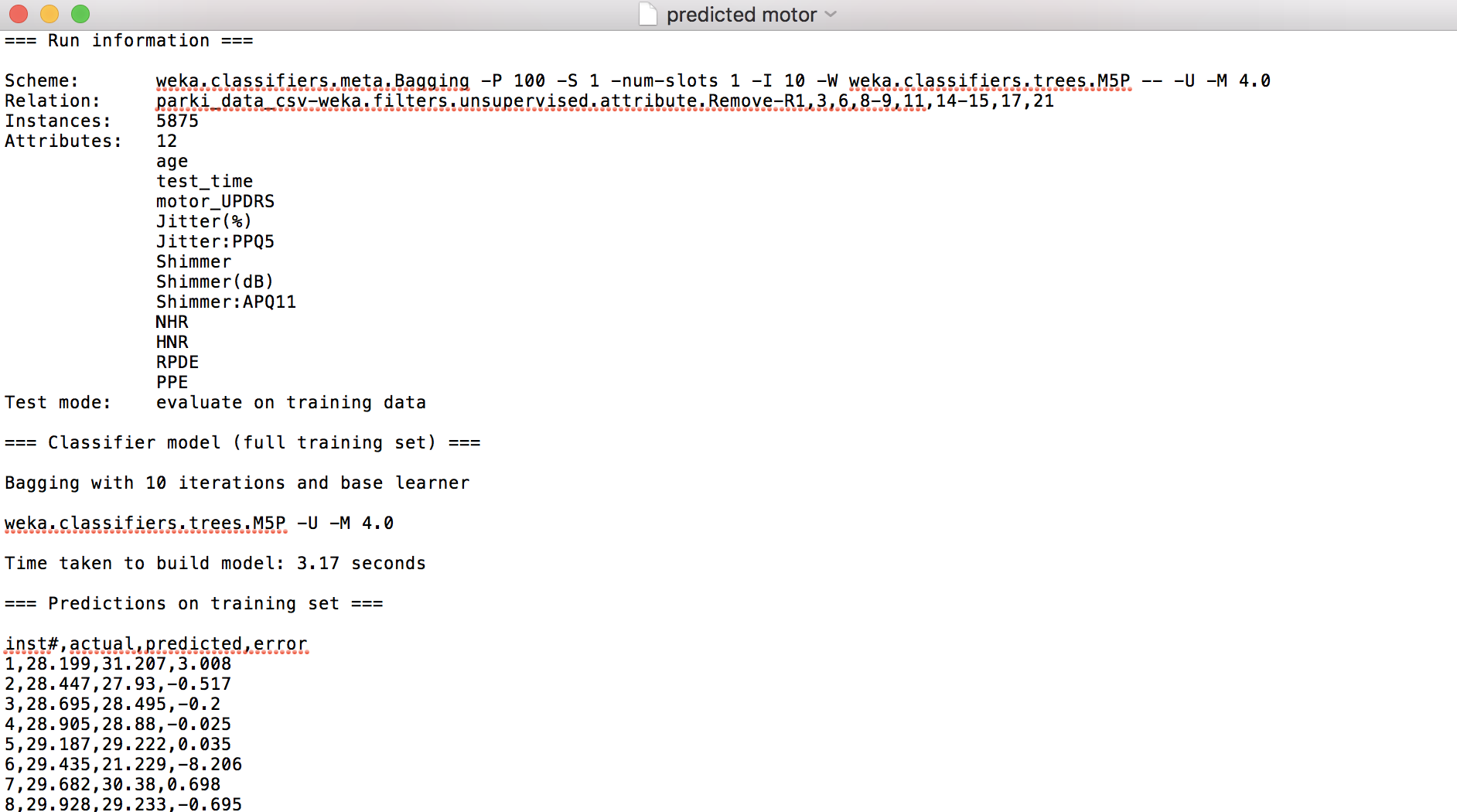
.

.

**D. M5P on test data for *total\_UPDRS***

****

**E. Predicted *total\_UPDRS* result**



.

.

.