PREDICTION PARKINSON'S DISEASE THROUGH VOICE MEAUSERS

Data Spaces

Tabriz Nuruyev
S263669@studenti.polito.it

1 Problem Introduction

Parkinson's disease is a progressive nervous system disorder that affects movement. Symptoms of the disease start gradually and can be different for everyone. Tremor, slow movements, speech changes are some of most common symptoms. While Parkinson's cannot be cured, early detection of disease can be crucial for starting appropriately targeted therapies immediately. In this research, different features extracted from patients' speech have been used as possible indicators of Parkinson's disease.

2 Dataset

For this problem, dataset provided by UCI Machine Learning Repository has been utilized. The dataset is consisted on 197 instances and 23 attributes. Each attribute is voice measure acquired from speech of patients. Finally, there is status column which is set to 0 when a patient is healthy and 1 when there is Parkinson's.

```
data=pd.read_csv(r"C:\Users\xazar\Downloads\parkinsons_data.txt")
data.head()

#scale is needed
```

	name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	...	Shir
0	phon_R01_S01_1	119.992	157.302	74.997	0.00784	0.00007	0.00370	0.00554	0.01109	0.04374	...	
1	phon_R01_S01_2	122.400	148.650	113.819	0.00968	0.00008	0.00465	0.00696	0.01394	0.06134	...	
2	phon_R01_S01_3	116.682	131.111	111.555	0.01050	0.00009	0.00544	0.00781	0.01633	0.05233	...	
3	phon_R01_S01_4	116.676	137.871	111.366	0.00997	0.00009	0.00502	0.00698	0.01505	0.05492	...	
4	phon_R01_S01_5	116.014	141.781	110.655	0.01284	0.00011	0.00655	0.00908	0.01966	0.06425	...	

5 rows × 24 columns

General view of dataset

3 Features and Processing

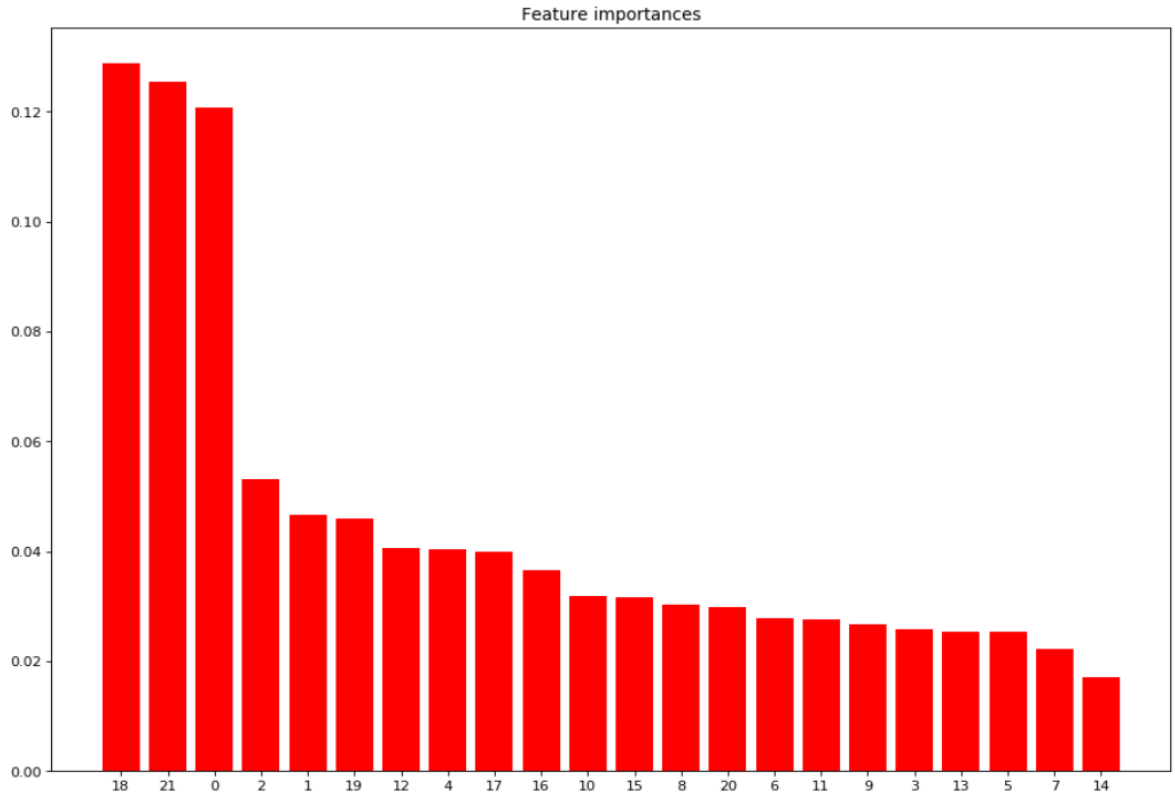
As seen above, features have different range in their values, therefore scaling should be applied to the features. In addition, "name" attribute can be dropped, as it will not provide any valuable information by intuition. The dataframe has also been checked for any missing value but there is none. As there are not many attributes, in main part of the research all features have been used in predictive model. In addition, two different feature selection algorithms still have been applied to dataset separately to check performance of the model. Firstly, recursive feature selection with cross-validation has been applied and it produced 8 features that it deemed more important. Support_ attribute of RFECV function provided "True" or "False" for each attribute in which "True" indicates importance of attribute.

[True False False True True False False True True False True False

True False False False False False True False False False]

Feature importance array by RFECV algorithm

Second method used is feature importance by forest of trees. Graph below visualizes attributes ranked by importance according to the forest of trees algorithm. Values in the x axis indicates sequence number of attributes. Results of these two methods discussed in result and discussion section.



Ranking of features by forest of trees algorithm

Finally, data has been split to train, validation and test portions with 5:3:2, respectively.

4 Evaluation Metrics

In this problem, main goal of the analysis is predicting the existence of disease. For such problems, evaluating performance of model by mere metric like accuracy does not provide much value. Furthermore, there is a considerable difference in number of instances belonging each class, so evaluating model by accuracy would yield misleading result. However, confusion matrix is widely used for medical application for a good reason, therefore it is deemed as more appropriate method. Especially in mentioned field, type 2 error is more serious problem than type 1 error. That is why in this paper, more emphasis has been put on recall rather than precision.

5 Models and Techniques

KNN – KNN is one of the most robust techniques for classification problem. If nearest neighbor parameter K chosen correctly, it can provide accurate results. In this paper, multiple nearest neighbor parameters have been checked to find optimal number for K. Furthermore, it is best practice to choose odd number for K to avoid ties in binary classification. That is why all the checked numbers are odd.

SVM – SVM is another widely used machine learning algorithm for binary classification. It tries to find optimal hyperplane to separate data points. In this algorithm, regularization parameter C and kernel coefficient γ need to be chosen carefully to acquire optimal results. For this reason, **Grid Search** have been used for tuning mentioned parameters.

6 Results and Discussion

KNN: Testing data split have been trained with nearest neighbor numbers of 1, 3, 5, 7. For obvious reasons, k=1 performed flawless in train data, but not so well in validation data. General performance drop has been noticed for values of nearest neighbor greater than 7 therefore they have not included in the report. Classification report for k=[1,3,5,7] is illustrated below:

```
Prediction accuracy for k= 1 :  0.775
Classification report for k= 1 :
              precision    recall  f1-score   support

           0       0.67      0.71      0.69        14
           1       0.84      0.81      0.82        26

    accuracy                           0.78        40
   macro avg       0.75      0.76      0.76        40
weighted avg       0.78      0.78      0.78        40

Prediction accuracy for k= 3 :  0.825
Classification report for k= 3 :
              precision    recall  f1-score   support

           0       0.82      0.64      0.72        14
           1       0.83      0.92      0.87        26

    accuracy                           0.82        40
   macro avg       0.82      0.78      0.80        40
weighted avg       0.82      0.82      0.82        40

Prediction accuracy for k= 5 :  0.8
Classification report for k= 5 :
              precision    recall  f1-score   support

           0       0.80      0.57      0.67        14
           1       0.80      0.92      0.86        26

    accuracy                           0.80        40
   macro avg       0.80      0.75      0.76        40
weighted avg       0.80      0.80      0.79        40

Prediction accuracy for k= 7 :  0.8
Classification report for k= 7 :
              precision    recall  f1-score   support

           0       0.88      0.50      0.64        14
           1       0.78      0.96      0.86        26

    accuracy                           0.80        40
   macro avg       0.83      0.73      0.75        40
weighted avg       0.81      0.80      0.78        40
```

Classification report for KNN algorithm

K=3 is picked as better parameter and applied to test data. In testing data, KNN performed even better with 91% accuracy overall and 1.00 recall for class label “1” which in terms means it detected all patients with Parkinson’s disease.

SVM: For finding optimal hyper-parameters, grid search has been applied to the training data and tested on validation data. It is found that C=1 and γ=0.1 are optimal parameters. Next, SVM algorithm is used with found parameters to predict testing data. It achieved 86% accuracy with 0.98 recall for class label “1”. For kernel, RBF has been used as it performed better than linear kernel.

For further analysis, train and validation data have been merged. Merged data has been trained by previous algorithms to check performance on test data. For KNN, results are even better. Again, k=3 is optimal value for number of nearest neighbors. The new model achieved 96.5 % accuracy and 1.00 recall. Graph below illustrates classification report of KNN algorithm with k=[1,3,5,7].

```
Prediction accuracy for k= 1 :  0.9655172413793104
Classification report for k= 1 :
              precision    recall  f1-score   support

           0       1.00      0.80      0.89        10
           1       0.96      1.00      0.98        48

    accuracy                           0.97        58
   macro avg       0.98      0.90      0.93        58
weighted avg       0.97      0.97      0.96        58

Prediction accuracy for k= 3 :  0.9655172413793104
Classification report for k= 3 :
              precision    recall  f1-score   support

           0       1.00      0.80      0.89        10
           1       0.96      1.00      0.98        48

    accuracy                           0.97        58
   macro avg       0.98      0.90      0.93        58
weighted avg       0.97      0.97      0.96        58

Prediction accuracy for k= 5 :  0.9482758620689655
Classification report for k= 5 :
              precision    recall  f1-score   support

           0       1.00      0.70      0.82        10
           1       0.94      1.00      0.97        48

    accuracy                           0.95        58
   macro avg       0.97      0.85      0.90        58
weighted avg       0.95      0.95      0.94        58

Prediction accuracy for k= 7 :  0.9482758620689655
Classification report for k= 7 :
              precision    recall  f1-score   support

           0       1.00      0.70      0.82        10
           1       0.94      1.00      0.97        48

    accuracy                           0.95        58
   macro avg       0.97      0.85      0.90        58
weighted avg       0.95      0.95      0.94        58
```

Classification report for KNN with merged data

However, when grid search applied to new merged data using SVM, change in optimal parameters has been noticed. While C=1 and γ=0.1 were deemed best hyper-parameters in previous application, with new training data, optimal values have been changed to C=100 and γ=0.1. Results of grid search illustrated below:

```
Detailed classification report:

The model is trained on the full development set.
The scores are computed on the full evaluation set.

              precision    recall  f1-score   support

           0       0.89      0.80      0.84        10
           1       0.96      0.98      0.97        48

    accuracy                           0.95        58
   macro avg       0.92      0.89      0.91        58
weighted avg       0.95      0.95      0.95        58
```

Grid Search

As seen in the graph, SVM reached 94.8% accuracy with 0.98 recall. As in previous application, KNN performed slightly better than SVM. General performance increase with new training data is understandable, as training data has been increased by 40% and it lead to better model generation.

As mentioned before, two feature selection method have been applied to dataset for further investigation. Original data consisted on 22 features after dropping status (which is label data) and name columns. Generally, feature selection or feature generation techniques are not applied to datasets with a few numbers of features, so it is done in this paper for solely experimental reasons. Same procedure has been applied to new data with selected features and results were not so surprising: any performance increase was not noticed.

8 Conclusion

In this paper, two robust machine learning algorithms have been used to correctly predict the existence of Parkinson's disease. Both algorithms produced accurate results, while KNN performed slightly better with respect to SVM. When training data have been increased, even better results have been produced by models. Especially, finding 1.00 recall for existence of PD is considerable result, because in medical applications, failing to detect existing disease is more serious problem than falsely assigning healthy patient with disease. As dataset has a few attributes, applying feature selection algorithms did not increase the overall performance of the model.

9 References

[1] Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection', Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM. BioMedical Engineering OnLine 2007, 6:23 (26 June 2007)

[2] Guo G., Wang H., Bell D., Bi Y., Greer K. (2003) KNN Model-Based Approach in Classification. In: Meersman R., Tari Z., Schmidt D.C. (eds) On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science, vol 2888. Springer, Berlin, Heidelberg

[3] Scikit-learn.org. 2020. *Sklearn.Svm.SVC — Scikit-Learn 0.22.2 Documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html> [Accessed 25 April 2020].