NEERAJ REDDY GUNDA

CS02505 – DATA MINING 1

CLASSIFICATION OF PARKINSON'S DISEASE USING CLASSIFICATION ANALYSIS

Project Report

TASK DESCRIPTION

Parkinson's Disease(PD) is a neurological disorder marked by Decreased dopamine levels in the brain. There is commonly observed effect on speech ,including difficulty in articulating sounds, lowered volume . Traditional diagnosis of Parkinson's Disease involves a clinician taking a neurological history of the patient and observing the sound frequencies in various situations. Since , there is no laboratory test for detecting PD, diagnosis is often difficult, particularly in early stages when there is no severve symtoms.here we diagnose disease using various data mining classification techniques.

There is currently no objective method for diagnosing PD. It can take months to get a reliable PD diagnosis, and symptoms need to be carefully monitored. Even then the probability of an inaccurate diagnosis is approximately 25%.

We used the bio medical voice measurements of 31 people.The results confirmed that voice measurements is relevant in diagnosing and monitoring PD.

DATA SET DESCRIPTION

The data set is obtained from uci repository. The data set consists of biomedical voice measurement of 31 people, 23 out of these 31 have PD, and the total recordings of these 31 people are represented as 195 rows. The original data set consists of 24 attributes. The 1st attributes is the name of the patient and the 18th attribute is the status of the patient which s 0 for a healthy patient and 1 for a patient with Parkinson's disease. The other attributes are the voice measures based on these voice measures the actually classification would take place.

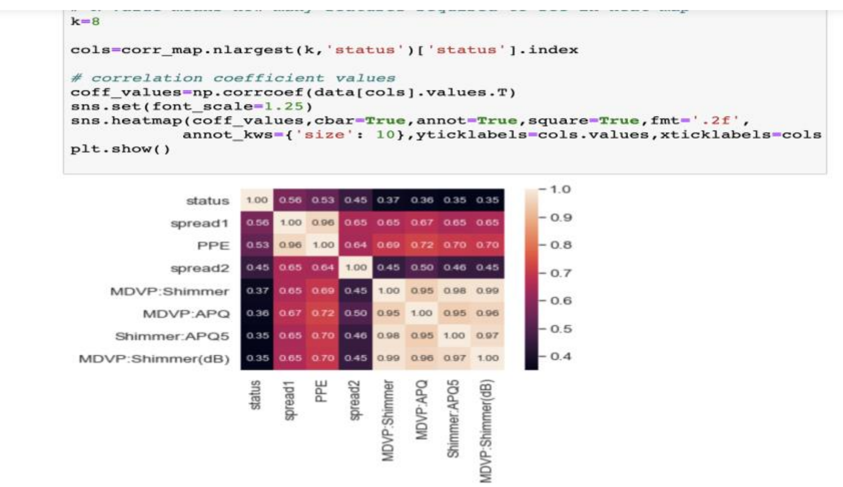
These include different types of fundamental frequencies, Jitter, shimmer, ratio of noise to tonal components, scaling components, and fundamental frequency measurements.

The data is passed over the data preprocessing phases e.g. data cleaning,missing values and transformed before applying four classification algorithms.

DATA PRE-PROCESSING

Histogram plot visualisation for each attribute will be so diffcult because we have high dimensional column. So Better, we can use heat map to find the correlations coefficient values and we can remove the irrelavant features it will minimize the accuracy of an algorithm.

we have correaltion values in each attribute in descending order so we are going to drop from MDVP:RAP column to MDVP:Fhi(Hz) because it have less correlation with other columns.



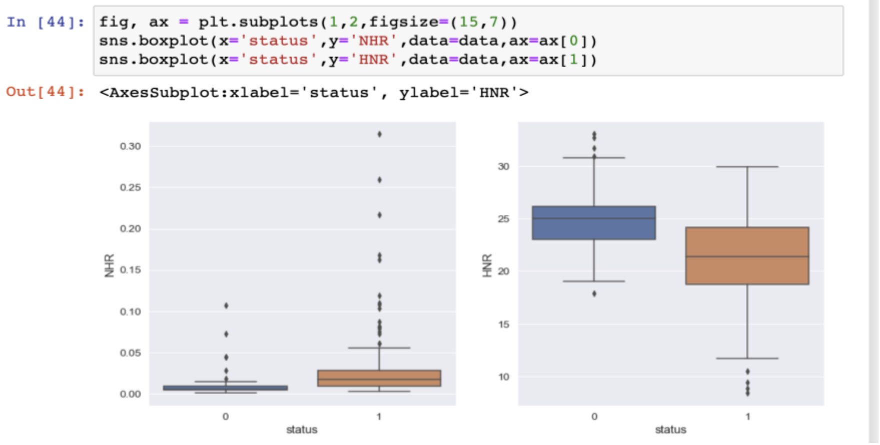
```
k=8

cols=corr_map.nlargest(k,'status')['status'].index

# correlation coefficient values
coff_values=np.corrcoef(data[cols].values.T)
sns.set(font_scale=1.25)
sns.heatmap(coff_values,cbar=True,annot=True,square=True,fmt='.2f',
            annot_kws={'size': 10},yticklabels=cols.values,xticklabels=cols
plt.show()
```

	status	spread1	PPE	spread2	MDVP:Shimmer	MDVP:APQ	Shimmer:APQ5	MDVP:Shimmer(dB)
status	1.00	0.56	0.53	0.45	0.37	0.36	0.35	0.35
spread1	0.56	1.00	0.96	0.65	0.65	0.67	0.65	0.65
PPE	0.53	0.96	1.00	0.64	0.69	0.72	0.70	0.70
spread2	0.45	0.65	0.64	1.00	0.45	0.50	0.46	0.45
MDVP:Shimmer	0.37	0.65	0.69	0.45	1.00	0.95	0.98	0.99
MDVP:APQ	0.36	0.67	0.72	0.50	0.95	1.00	0.95	0.96
Shimmer:APQ5	0.35	0.65	0.70	0.46	0.98	0.95	1.00	0.97
MDVP:Shimmer(dB)	0.35	0.65	0.70	0.45	0.99	0.96	0.97	1.00

#k = 8



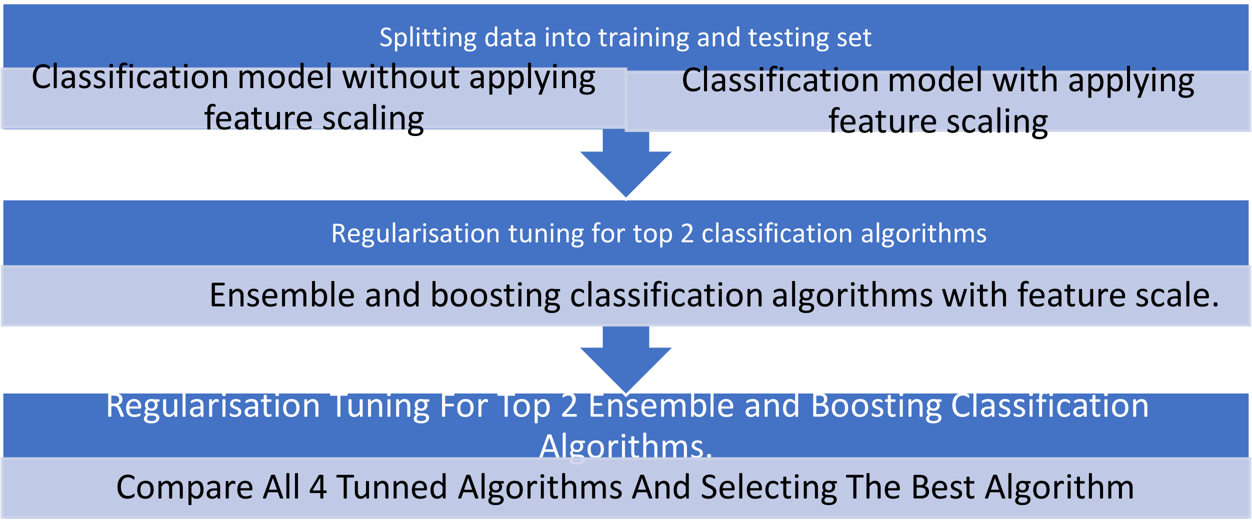
```
In [44]: fig, ax = plt.subplots(1,2,figsize=(15,7))
         sns.boxplot(x='status',y='NHR',data=data,ax=ax[0])
         sns.boxplot(x='status',y='HNR',data=data,ax=ax[1])

Out[44]: <AxesSubplot:xlabel='status', ylabel='HNR'>
```

People who have PD(status equal to one) have higher levels of outliers. Also, looking into the HNR ratio people who have PD have lower levels in the same. Noise to Harmonic Ratio (NHR) is another useful measure . This can be routinely measured using MDVP. For a signal that can be assumed to be periodic (vowels), the signal-to-noise ratio will be equal to the harmonics-to-noise ratio (HNR)

CLASSIFIERS USED AND RESULTS

- **DECISION TREE**
- **RANDOM FOREST**
- **K NEAREST NEIGHBOUR**
- **LOGISTIC REGRESSION**


Splitting data into training and testing set
Classification model without applying feature scaling
Classification model with applying feature scaling
Regularisation tuning for top 2 classification algorithms
Ensemble and boosting classification algorithms with feature scale.
Regularisation Tuning For Top 2 Ensemble and Boosting Classification Algorithms.
Compare All 4 Tunned Algorithms And Selecting The Best Algorithm

APPLYING CLASSIFICATION WITHOUT FEATURE SCALING:

LOGISTIC REGRESSION:

It uses the odds of correct diagnosis of parkinsons disease. The study uses status variable as a dependent variable and comp 1 and comp 4 as the covariates of to construct the logistic model,it uses the p value and chi square value.

By using the logistic regression without applying feature scaling we got an accuracy of 0.86044.

RANDOM FOREST

It works by generating several decision tree and combining their result for classification.it isbasically a bagging idea for random feature selector. Here no of tree = 3 and k = 10.

By using random forest we got an accuracy of 0.897802.this performed best among other classifier

K – NEAREST NEIGHBOUR

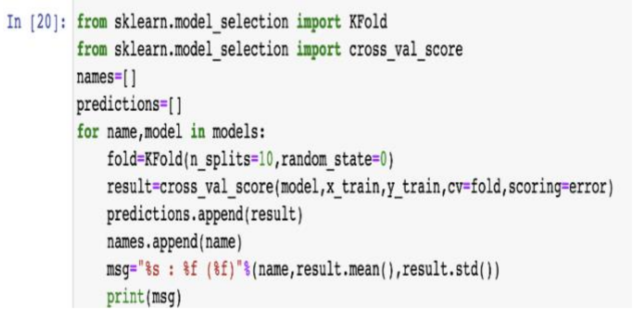
It is a lazy approach that classifies an instance based upon knumber of closest neighbours,k was selected as 5 distance matrix was Manhattan and k – fold validation was k = 10.

DECISION TREE

In the DT healthy status is response variable and the 22 voice variables are input variable to the tree. 22 of 24 variable because we dropped two variables

LOGISTIC REGRESSION	**0.860440 (0.089956)**
KNN	0.808242 (0.119180)
RANDOM CLASSIFIER	0.897802 (0.046115)
DECISION TREE	0.852198 (0.048117)

CLASSIFICATION WITH FEATURE SCALING



```
In [20]: from sklearn.model_selection import KFold
         from sklearn.model_selection import cross_val_score
         names=[]
         predictions=[]
         for name,model in models:
             fold=KFold(n_splits=10,random_state=0)
             result=cross_val_score(model,x_train,y_train,cv=fold,scoring=error)
             predictions.append(result)
             names.append(name)
             msg="%s : %f (%f)"%(name,result.mean(),result.std())
             print(msg)
```

- In feature scaling we used the same algorithms to calculate the accuracies.for scaling we used **MIN-MAX SCALER** this estimates and translates each feature ,such that it is in given range on training set between o and 1.
- By using feature scaling we got almost similar accuracies there is a slight variation in random forest classifier.most of the times our dataset will contain features with high values ,magnitude and range.
- Tree based models are not distance bnased models varyiong range of features

LOGISTIC REGRESSION	**0.860440 (0.089956)**
KNN	0.808242 (0.119180)
RANDOM FOREST	0.912088 (0.052930)
DECISION TREE	0.837912 (0.063330)

TUNING FOR TOP 2 CLASSIFICATION ALGORITHMS

Tuning is nothing but changing the hyper parameters .i used four parameters we can use many but it increases the time complexity.

- grid=GridSearchCV(estimator=model,param_grid=param_grid,scoring=error,cv=fold)
- Gridsearchcv considers all the parameter combinations.
- By applying Tuning we got random tree and logistic regression as the best
  Best : 0.861bfor logistic regression and best : 0.897 for random forest

ENSEMBLE AND BOOSTING CLASSIFICATION ALGORITHMS

The goal of the ensemble methods is to combine the predictions of several base estimators built with a given learning algorithms in order to improve the robustness over single estimator.
Bagging builds several estimators independently and then average the predictions in these algorithms random forest gives the best performance.
we used pipeline to assemble several steps that can be cross validated together while setting different parameters.

ADA BOOST	**0.839560 (0.081896)**
GRADIENT BOOSTING	**0.867582 (0.070383)**
RANDOM FOREST	**0.912088 (0.065852)**

COMPARE ALL 4 TUNNED ALGORITHMS AND SELECTING THE BEST

we got testing set accuracy of 91.20% by using random forest classifier.

train set 1.0

train set matrix [[32 0] [0 104]]

test set matrix [[11 5] [0 43]]

Gradientboostclassifier	**Best: 0.897802 using {'n_estimators': 200}**
Random forest classifier	Best: 0.912088 using {'learning_rate': 0.7, 'n_estimators': 100}
LogisticRegression	0.860440 (0.089956)
decision_tree	0.837912 (0.063330)

CONCLUSION

By data preprocessing especially irrelevant dimensions and outliers are removed. Once the data is similar to the normal distribution four methods are used to classify the data. K-NN, Random Forest, and Gradientboost ,decision tree are used. The comparative analysis shows that out of the four Random Forest is the best model for classification with accuracy of 91.20 using k=10 fold validation and 70: 30 split. The model was not underfitted or overfit it fitted perfectly .

The study uses the data mining analysis to explore the Parkinson's Disease data. Data mining is widely used in the realm of the preventive medicine. By means of the study of the PD data, medical researchers can create the evaluation table according to the results of data mining in order to make physicians and ordinary people aware the early symptoms of PD and make earlier treatments.