**Machine Learning in Healthcare & Biomedicine Module Assignment**

**Introduction**

**Background**

The liver is an organ present only in vertebrates, detoxifies various metabolites, synthesis, synthesizes proteins, and produces biochemicals necessary for digestion (1, 2). Liver disease is hard to be diagnosed in an early stage as the organ will be functioning normally even when it is partially damaged. An early diagnosis of liver issues with laboratory analysis of a range of proteins in the bloodstream will increase patients survival rate and reduce the waiting times to access medical care.

In chosen dataset each instance represents a patient with a range of medical tests known to assess the onset of a broad range of liver diseases, some of the most prominent ones being presented in Table 1 (2).

Table 1: Most frequent liver diseases.

|  |  |
| --- | --- |
| **Condition** | **Possible causes** |
| Alcohol-related liver disease | Regularly drinking too much alcohol |
| Non-alcoholic fatty liver disease | Being very overweight (obese) – this may cause fat to build up in the liver |
| Hepatitis | Catching a viral infection, regularly drinking too much alcohol |
| Haemochromatosis | A gene that runs in families and may be passed from parents to children |
| Primary biliary cirrhosis | May be caused by a problem with the immune system |

The data was recorded in the Andhra Pradesh state of India (3, 4). Contains 583 instances with 11 features, including the Class feature to be predicted. It contains 441 male patient records and 142 female patient records (Fig. 1). In Table 2 are shown the descriptive statistics of the dataset. The average instance has an average of 45 years old and is of male gender. Apparently, maximum values for Direct Bilirubin, Alkaline Phosphatase, SGPT and SGOT are outliers in the dataset, probably because of data recording or transcribing, since their values are more than 3 standard deviations from the mean. The dataset contains 416 liver patient records and 167 non liver patient records, showing a heavy skew towards the positive class (Fig. 1).

**Exploratory Data Analysis**

For most readings there is a normal distribution, as can be seen in figure 1, and as explained before, there are outliers that could be interfering with the distribution. Features Alkaline Phosphatase, Direct Bilirubin, Total Bilirubin, SGOT and SGPT appear to have a Poisson distribution. The correlation matrix shows how each feature correlates with the onset of liver diseases. Although some correlations are distinguishable, these are not particularly strong. It’s possible to highlight the relationship between Direct Bilirubin (Fig. 2 and 3) readings with the liver patient class, as well as Alkaline Phosphatase levels. Total Protein readings don’t correlate with the positive class (Fig. 2 and 4), which suggests that the amount of protein in blood plasma doesn’t vary with liver disease, unlike those of Alkaline Phosphatase, SGPT and SGOT. The proportion of these proteins could be of importance, and are added to the original features as described in the feature engineering section. Also, there is no correlation between the disease and gender, but there is a small positive correlation to age.

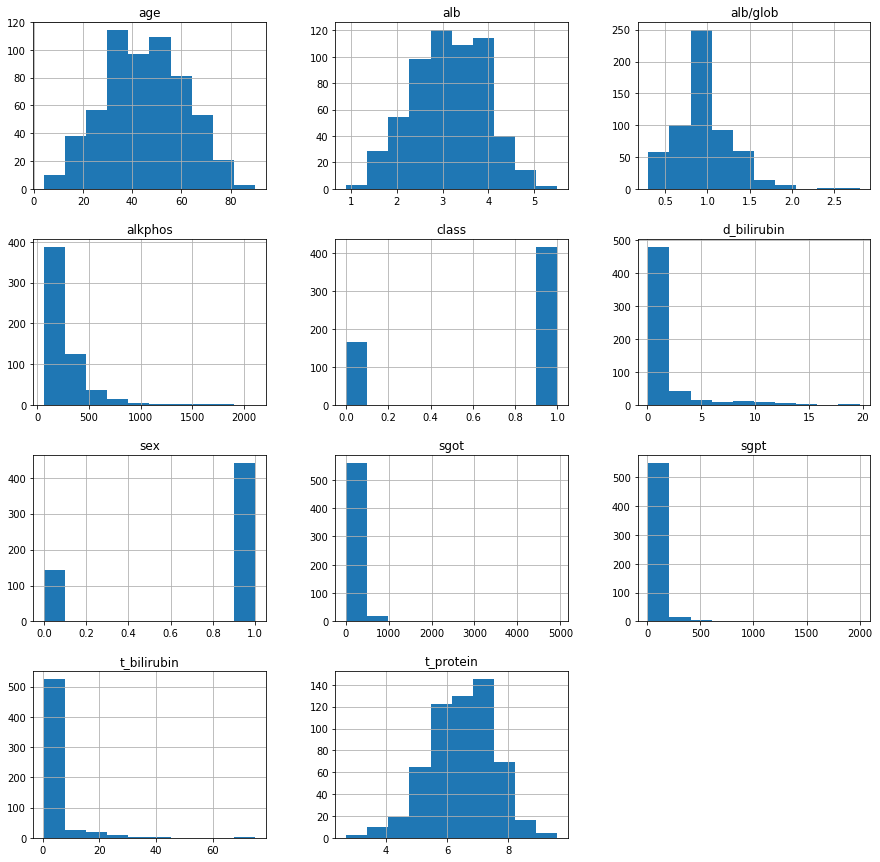
****

Figure 1: Distribution of readings in the Indian Liver Patient Dataset

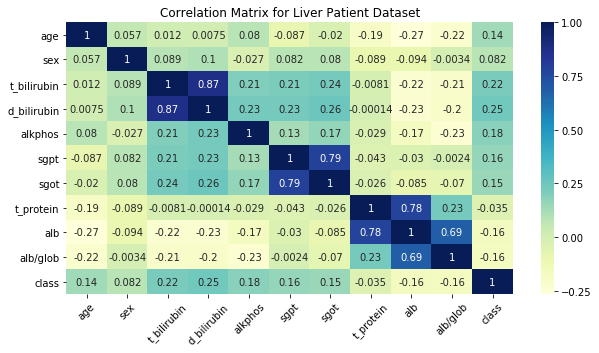
Table 2: descriptive statistics of the Indian Liver Patient Dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Age** | **Sex** | **Total Bilirubin** | **Direct Bilirubin** | **Alkaline Phosphatase** | **SGPT** | **SGOT** | **Total Protein** | **Albumin** | **Albumin/Globulin** | **Class** |
| **Count** | 583 | 583 | 583 | 583 | 583 | 583 | 583 | 583 | 583 | 583 | 583 |
| **Mean** | 44.74 | 0.75 | 3.29 | 1.49 | 290.58 | 80.71 | 109.91 | 6.48 | 3.14 | 0.95 | 0.71 |
| **Std** | 16.18 | 0.42 | 6.20 | 2.81 | 242.94 | 182.62 | 288.92 | 1.09 | 0.80 | 0.32 | 0.45 |
| **Min** | 4 | 0 | 0.4 | 0.1 | 63 | 10 | 10 | 2.7 | 0.9 | 0.3 | 0 |
| **25.00%** | 33 | 1 | 0.8 | 0.2 | 175.5 | 23 | 25 | 5.8 | 2.6 | 0.7 | 0 |
| **50.00%** | 45 | 1 | 1 | 0.3 | 208 | 35 | 42 | 6.6 | 3.1 | 0.93 | 1 |
| **75.00%** | 58 | 1 | 2.6 | 1.3 | 298 | 60.5 | 87 | 7.2 | 3.8 | 1.1 | 1 |
| **Max** | 90 | 1 | 75 | 19.7 | 2110 | 2000 | 4929 | 9.6 | 5.5 | 2.8 | 1 |

Table 3: Features of the Indian Liver Patient Disease Dataset. No measurement units are provided.

|  |  |  |
| --- | --- | --- |
| **Features** | **Type** | **Description** |
| Gender | Categorical | Patient’s Gender |
| Age | Real number | Patient’s Age |
| TB | Real number | Total Bilirubin |
| DB | Real number | Direct Bilirubin |
| TP | Real number | Total Protein |
| ALB | Real number | Albumin |
| A / G ratio | Real number | Albumin to Globulin ratio |
| SGPT | Integer | Alanine transaminase (formerly **S**erum **G**lutamate-**P**yruvate **T**ransaminase) |
| SGOT | Integer | Aspartate transaminase (formerly **S**erum **G**lutamic **O**xaloacetic **T**ransaminase) |
| Alkphos | Integer | Alkaline Phosphatase |
| Class | Integer | Selector field used to split the data into two sets (labelled by the experts) |

The positive correlation can be seen between the positive class and other laboratory tests like Direct Bilirubin (Fig. 2), where the readings for Direct Bilirubin SGPT and SGOT (Figures 3, 5 and 6).

Figure 2: Correlation Matrix of the Indian Liver Patient Dataset.

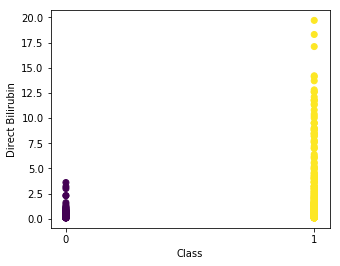
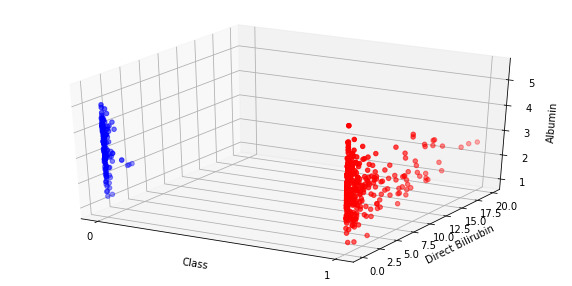
****

Figure 3: Distribution of Direct Bilirubin readings in each Class.



Figure 4: Distribution of Total Protein readings in each Class.

Figure 5: Distribution of Direct Bilirubin (positive correlation with class) and Albumin (no significant correlation with class) readings in each Class.

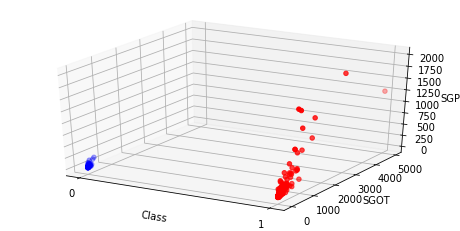
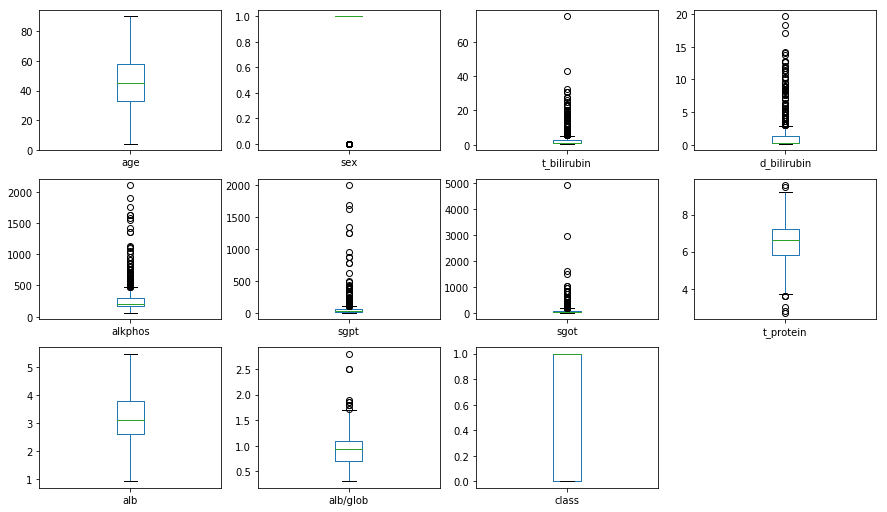


Figure 6: Distribution of SGOT (positive correlation with class) and SGPT (positive correlation with class) readings in each Class.

The box plots (Fig. 7) show a better display of the sample distribution in the dataset, since a number of samples fall outside the whiskers of the plot. Some of these outliers are treated in the following section.

Figure 7: Distribution box plots for the features of the Indian Liver Patient Dataset.

**Methods**

**Data Cleansing & Preprocessing**

The dataset needed to be processed before submiting to a machine learning algorithm. The categorical Sex feature was either a “male” or “female” string, which was mapped to 1 or 0, respectively. The Class feature was 1 for liver patients and 2 for non-patients, the latter being changed to 0. Regarding missing values, the Albumin/Globulin ratio had 4 NA values, which were imputed with the median value of the corresponding feature. Instances with outliers in their features values were removed from the dataset if their z-score values were bigger than 3, which resulted in a final dataset with 536 instances.

**Feature Engineering**

New features were created which could be of importance for a classification algorithm (Table 4). As seen in the correlation plot above (Fig. 1), there is no correlation between Class and Total Protein, but proportion of measured proteins could be relevant for disease detection.

Table 4: Features created for the Indian Liver Patient Dataset.

|  |  |
| --- | --- |
| **Feature created** | **Formula** |
| Indirect Bilirubin (5) | ind\_bilirubin = t\_bilirubin - d\_bilirubin |
| Proportion of SGPT in Total Protein | sgpt\_prop = sgpt / t\_protein |
| Proportion of SGOT in Total Protein | sgot\_prop = sgot / t\_protein |
| Proportion of SGPT and SGOT in Total Protein | sgot\_sgpt\_prop = (sgpt + sgot) / t\_prot |
| Proportion of Alkaline Phosphatase in Total Protein | alkphos\_prop = alkphos / t\_protein |
| Globulin (5) | glob = alb / (alb/glob) |
| Proportion of Globulin in Total Protein | glob\_prop = glob / t\_protein |
| Proportion of Albumin in Total Protein | alb\_prop = alb / t\_protein |

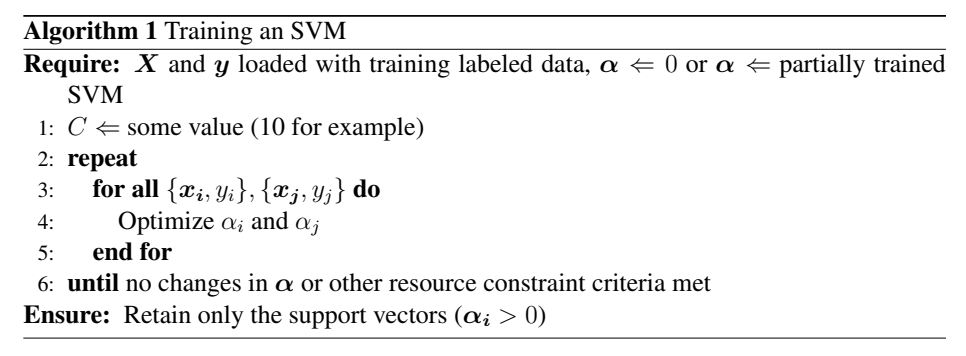
The data was split into training and test sets, the latter being 15% of the data. Class imbalance was addressed oversampling the minority class with the SMOTE algorithm (6) to a ratio of 1:1. Then, a scaling was performed first on the training set, and subsequently on the testing data to avoid data leakage.

Feature selection was done independently for the base Support Vector Machine and Random Forest algorithms with the Sequential Backward Selection function provided in the practical lectures, according to which feature combination results in the best Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) score (Tables 5 and 8).

Optimization of hyper-parameters for both algorithms was performed with a series of applications of the Gridsearch-CV function with 10 fold Cross-validations of the training dataset, basing the scoring criteria on the ROC-AUC score, starting with a broad range of increasing values for the hyper-parameters (Tables 11 and 16).

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

The SVM is a binary classification algorithm where input vectors are non-linearly mapped to a high dimension feature space where a linear decision surface (plane, or hyperplane in higher dimensions) is constructed, which properties ensures the high generalization ability of the algorithm at the time of separating the two classes (7), the support vectors being the data points that lie closest to said surface and the most difficult to classify. Ideally, new data points to be classified would fall between these support vectors and the bulk of the training data. A pseudocode for the SVM is shown in fig. 8 (11).

Figure 8: Support Vector Machine pseudocode. *αi* and *αj*

are Lagrange multipliers.

*Base SVM*

The base SVM was fitted to the selected training data before optimization (Tables 5 and 6). Evaluation criteria on both datasets show a performance near random guessing. The Recall criteria is slightly better than random at 52.87% for the training data, and below 47% for the testing data. The Confusion Matrix (Table 7), has as a high number of False Negatives for both sets, being higher than the True Positive count for the testing data, and almost to the same count on the training data.

Table 5: Features selected for Support Vector Machine algorithm by Sequential Backward Selection according to the best ROC-AUC. Created features are denoted in bold.

|  |  |
| --- | --- |
| **Feature selected** | **Description** |
| age | Patient’s Age |
| sgpt | Alanine transaminase (formerly **S**erum **G**lutamate-**P**yruvate **T**ransaminase) |
| sgot | Aspartate transaminase (formerly **S**erum **G**lutamic **O**xaloacetic **T**ransaminase) |
| alb/glob | Albumin to Globulin ratio |
| **sgot\_prop** | **Aspartate transaminase proportion of Total protein** |
| **alb\_prop** | **Albumin proportion of Total Protein** |

It’s possible the selected features present little variance, making it difficult for the algorithm to fit properly, which can also be seen in the ROC curves based on the predicted probabilities, where it shows that performs slightly better than random guessing in the test data (Fig. 9).

Table 6: Evaluation criteria for the Base SVM algorithm

|  |  |  |
| --- | --- | --- |
|  | **Evaluation Criteria** | **Score** |
| **Training data** | Accuracy | 0.7134 |
| ROC-AUC score (prediction) | 0.7134 |
| Precision | 0.8384 |
| Recall | 0.5287 |
| F1 | 0.6484 |
| Error rate | 0.2866 |
| **Testing data** | Accuracy | 0.5556 |
| ROC-AUC score (prediction) | 0.6121 |
| Precision | 0.8125 |
| Recall | 0.4643 |
| F1 | 0.5909 |
| Error rate | 0.4444 |

Table 7: Confusion Matrix for the Base SVM for training and testing data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Predicted Values | |
|  |  |  | Negative | Positive |
| Actual Values | **Training data** | Negative | TN: 282 | FP: 32 |
| Positive | FN: 148 | TP: 166 |
| **Testing data** | Negative | TN: 19 | FP: 6 |
| Positive | FN: 30 | TP: 26 |

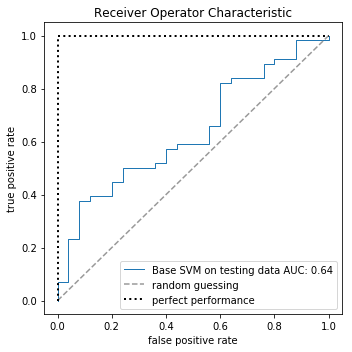
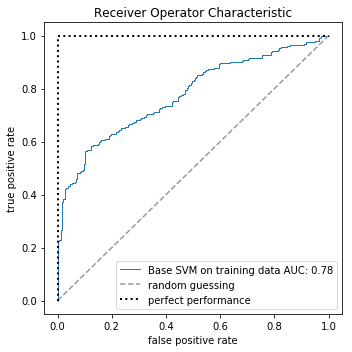
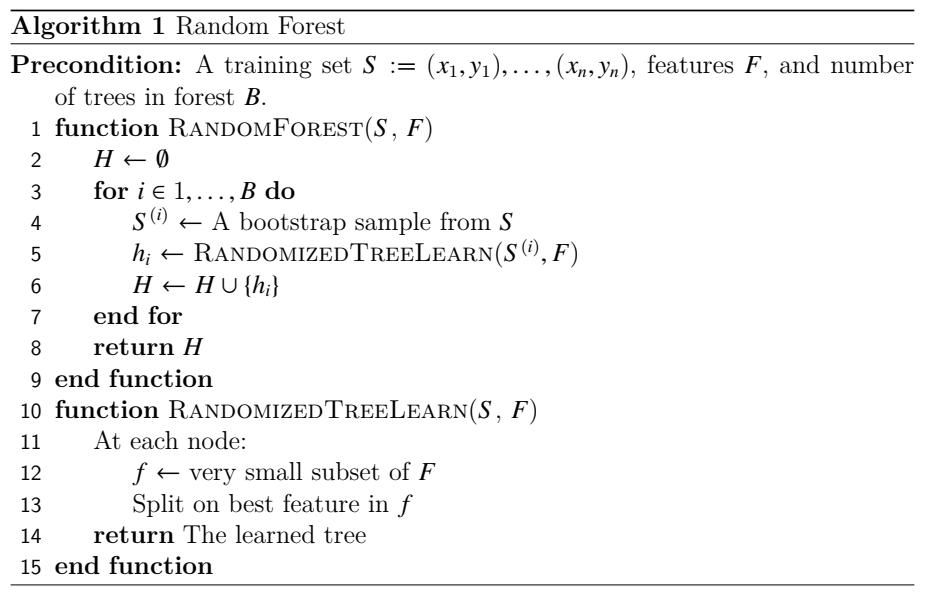


Figure 9: ROC curves with AUC for the Base SVM algorithm on the training and test datasets.

***Random Forest (RF)***

The RF classification algorithm is an ensemble of a given number of Decision Trees, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. (8). This results in a greater tree diversity, which trades a higher bias for a lower variance and generally gives better results than any individual Decision Tree. The randomness in building each tree of the RF forces the algorithm to consider many possible explanations, the result being that the RF captures a much broader picture of the data than a single tree, avoiding over fitting to the data and achieving a good generalization. A pseudocode of the RF algorithm is presented in fig. 10 (12).

 Figure 10: Random Forest Pseudocode

The base RF algorithm on the selected features was used as benchmark (Tables 8 and 9), resulting in high scores for the training data but not for the testing data.

Table 8: Features selected for the Random Forest algorithm. Selected created features are denoted in bold.

|  |  |
| --- | --- |
| **Feature selected** | **Description** |
| age | Patient’s Age |
| sex | Patient’s Sex |
| **ind\_bilirubin** | **Indirect Bilirubin** |
| **glob\_prop** | **Globulin proportion of Total Protein** |
| t\_bilirubin | Total Bilirubin |
| **sgpt\_prop** | **Alanine transaminase proportion of Total Protein** |
| d\_bilirubin | Direct Bilirubin |
| **sgot\_prop** | **Aspartate transaminase proportion of Total protein** |
| alkphos | Alkaline Phosphatase |
| **sgot\_sgpt\_prop** | **SGPT and SGOT proportion of Total Protein** |
| sgpt | Alanine transaminase (formerly **S**erum **G**lutamate-**P**yruvate **T**ransaminase) |
| sgot | Aspartate transaminase (formerly **S**erum **G**lutamic **O**xaloacetic **T**ransaminase) |
| alb/glob | Albumin to Globulin ratio |
| **alkphos\_prop** | **Alkaline Phosphatase proportion of Total Protein** |

The confusion matrix (Table 10) shows high numbers of False Negatives, reaching counts of almost 50% of the True Positives in the testing data. For the training, the counts for each category are in line with an overfitting of the data, as expected. The ROC curve and AUC shows little generalization capabilities for the RF (Fig. 11).

Table 9: Evaluation criteria for the Base Random Forest for training and testing data

|  |  |  |
| --- | --- | --- |
|  | **Evaluation Criteria** | **Score** |
| Training data | Accuracy | 0.9857 |
| ROC-AUC score (prediction) | 0.9857 |
| Precision | 0.9935 |
| Recall | 0.9777 |
| F1 | 0.9856 |
| Error rate | 0.0143 |
| Testing data | Accuracy | 0.6420 |
| ROC-AUC score (prediction) | 0.6193 |
| Precision | 0.7755 |
| Recall | 0.6786 |
| F1 | 0.7238 |
| Error rate | 0.3580 |

Table 10: Confusion Matrix for the Base Random Forest for training and testing data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Predicted Values | |
|  |  |  | Negative | Positive |
| Actual Values | **Training data** | Negative | TN: 312 | FP: 2 |
| Positive | FN: 7 | TP: 307 |
| **Testing data** | Negative | TN: 14 | FP: 11 |
| Positive | FN: 18 | TP: 38 |

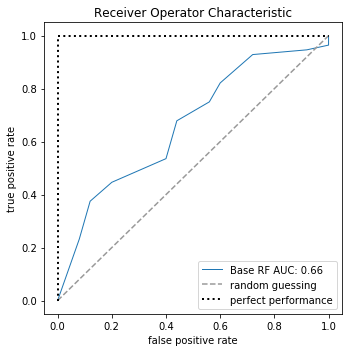
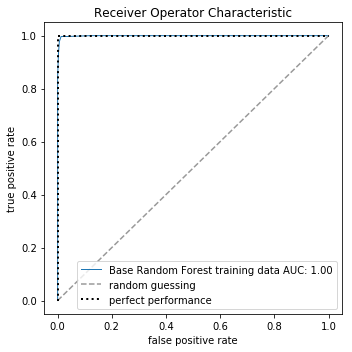
**

Figure 11: ROC curves with AUC for the base Random Forest algorithm on the training and test datasets.

**Results**

*Support Vector Machine*

The optimized hyper-parameters for the SVM algorithm are presented in Table 11. These did not show over fitting to the data, since the main evaluation criteria on the test set did not get worse during the optimization steps.

Table 11: Optimized hyper-parameters for the Support Vector Machine Algorithm

|  |  |  |
| --- | --- | --- |
| **Hyper-parameter** | **Function** | **Value** |
| Kernel | Similarity function: Gaussian Radial Basis Function | RBF |
| C | Penalty parameter C of the error term | 1.9 |
| decision\_function\_shape | Decision function | One-vs-One (ovo) |
| gamma | Kernel coefficient for ‘rbf’, controls the “narrowness” of the bell curve. | 2.3 |
| shrinking | Whether to use the shrinking heuristic | TRUE |

Evaluation criteria for the optimized SVM are presented in Table 12. Interestingly, the results with the test data set did not improve in a substantial manner when compared to the initial results with the base SVM algorithm. This suggests that a SVM is not an appropriate classification algorithm for the provided data set.

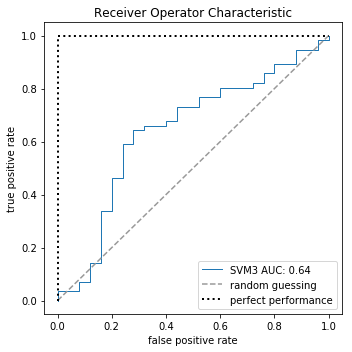
Table 12: Evaluation criteria for the optmized SVM algorithm.

|  |  |
| --- | --- |
| **Evaluation Criteria** | **Score** |
| Accuracy | 0.6420 |
| ROC-AUC score (prediction) | 0.6636 |
| Precision | 0.8293 |
| Recall | 0.6071 |
| F1 | 0.7010 |
| Error rate | 0.3580 |

The Confusion Matrix denotes a high number of False Negatives, reaching a count of more than half of the True Positives, which is worrisome in a medical context since the number of misdiagnosed patients would be outside any tolerable limits (Table 13).This suggests there is low uniformity in the positive class and the SVM algorithm is not able to classify both classes with the accuracy necessary to be used in a real setting. At the same time, the ROC curve changes its shape, but the AUC doesn’t improve its value (Fig. 12).

Table 13: Confusion Matrix of the optimized SVM algorithm**.**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted Values** | |
|  |  | Negative | Positive |
| **Actual Values** | Negative | TN: 18 | FP: 7 |
| Positive | FN: 22 | TP: 34 |

Figure 12: ROC curves with AUC for the optimized SVM algorithm

**Sensitivity analysis**

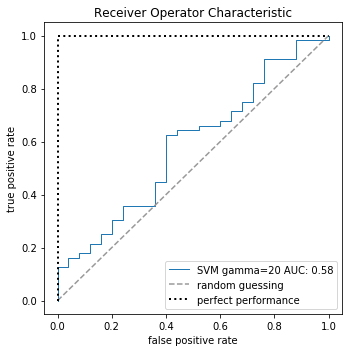
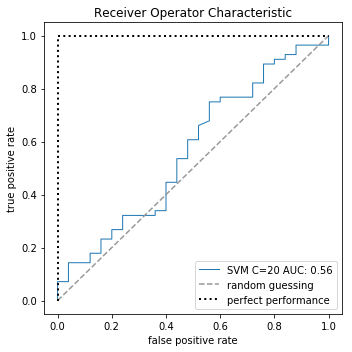
A sensitivity analysis on the optimized SVM, modifying the parameter C from 1.9 to 20, and gamma from 2.3 to 20, shows the deterioration of the evaluation criteria considered relevant (Tables 14 and 15. Fig. 13).

Table 14: Evaluation criteria for the sensitivity analysis for SVM.

|  |  |  |
| --- | --- | --- |
| **Evaluation Criteria** | **Score (C=20)** | **Score (gamma=20)** |
| Accuracy | 0.5802 | 0.6173 |
| ROC-AUC score (prediction) | 0.5525 | 0.5350 |
| Precision | 0.7292 | 0.7119 |
| Recall | 0.6250 | 0.7500 |
| F1 | 0.6731 | 0.7304 |
| Error rate | 0.4197 | 0.3827 |

Table 15: Confusion Matrix for the optimized SVM at C = 20 and gamma = 20.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Predicted Values** | | | |
|  |  | **C=20** | | **gamma=20** | |
|  |  | **Negative** | **Positive** | **Negative** | **Positive** |
| **Actual Values** | Negative | TN:12 | FP:13 | TN:8 | FP:17 |
| Positive | FN:21 | TP:35 | FN: 14 | TP:42 |

Figure 13: ROC-AUC at different values of C and gamma for the optimized SVM.

Modifying the C and gamma parameters to a higher order of magnitude shows how sensible the SVM algorithm is to its changing values. Although the Recall criteria improved with the higher values, every other criteria tracked showed lower scores. If the Recall criteria is taken as the primary evaluation parameter, paying attention to False negatives, it could be argued that an optimized SVM for ROC-AUC is not ideal method, and maximizing Recall would be a better optimization instead of the former. These results of the suggests that the SVM is more sensible to changes in orders of magnitude of its C parameter than the gamma value.

*Random Forest*

The optimized hyper-parameters for the RF algorithm are presented in Table 16. In this case, the optimization steps did show an over fitting to the data, since ROC-AUC score used to track the estimator in the grid search with cross-validation kept giving better results in each iteration but the evaluation criteria for the testing set reached a maximum and then started to deteriorate. The optimization was stopped at this point and the hyper-parameters chosen are presented.

Table 16: Optimized hyper-parameters for the Random Forest algorithm

|  |  |  |
| --- | --- | --- |
| **Hyper-parameter** | **Function** | **Value** |
| n\_estimators | Number of trees to be created | 1000 |
| criterion | The function to measure the quality of a split | Entropy |
| max\_depth | The maximum depth of the tree | 15 |
| max\_features | The number of features to consider when looking for the best split | Auto |
| min\_samples\_leaf | The minimum number of samples required to be at a leaf node | 1 |
| min\_samples\_split | The minimum number of samples required to split an internal node | 2 |
| class\_weight | Weights associated with classes | Balanced Subsample |
| oob\_score | Whether to use out-of-bag samples to estimate the generalization accuracy | True |
| warm\_start | To reuse the solution of the previous call to fit and add more estimators or start a new tree | True |

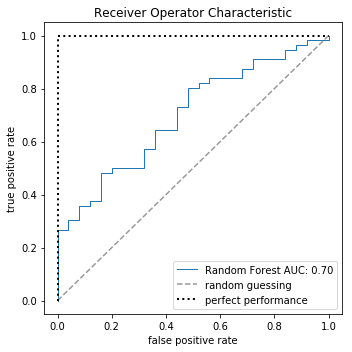
The evaluation criteria for the optimized RF with the test data set show an improvement with respect to the initial results with the base RF algorithm (Tables 17 and 18. Fig. 14). This suggests that it’s RF is at least a better classification algorithm than a SVM for the provided data set.

Table 17: Evaluation criteria for the optimized Random Forest.

|  |  |
| --- | --- |
| **Evaluation Criteria** | **Score** |
| Accuracy | 0.6790 |
| ROC-AUC score (prediction) | 0.6461 |
| Precision | 0.7885 |
| Recall | 0.7321 |
| F1 | 0.7593 |
| Error rate | 0.3210 |

Table 18: Confusion Matrix of the optimized Random Forest algorithm

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted Values** | |
|  |  | Negative | Positive |
| **Actual Values** | Negative | TN: 14 | FP: 11 |
| Positive | FN: 15 | TP: 41 |

Figure 14: ROC curve with AUC for the optimized Random Forest algorithm

**Sensitivity analysyis**

Modifying parameters corresponding to the number of trees per RF and the max depth of a tree to a lower order of magnitude shows how impactful these factors are when creating a RF algorithm. Diminishing the number of trees has impact on the robustness of the model, since there are less estimators with variability to create the ensemble. This is reflected in how every criteria deteriorates with a smaller number of trees (Table 19). The confusion matrix (Table 20) shows one more sample was added to the False Negatives group, giving a worse error rate. Interestingly, the ROC-AUC slightly improves to 0.71 (Fig. 15). This could mean that probabilities get slightly better, but since the test set has a small number of samples it could be factor affecting how the criteria is calculated.

Changing the max depth of the trees worsens all criteria but Precision, which reaches 80%. Nevertheless, in a medical context this wouldn’t be so critical, since non-patients being diagnosed as positive wouldn’t be at risk. The Recall remains remains low, as the so do the other criteria.

Table 19: Evaluation criteria for the sensitivity analysis for Random Forest.

|  |  |  |
| --- | --- | --- |
| **Evaluation Criteria** | **Score (number of trees = 100)** | **Score (max\_depth = 5)** |
| Accuracy | 0.6667 | 0.6420 |
| ROC-AUC score (prediction) | 0.6371 | 0.6414 |
| Precision | 0.7843 | 0.8000 |
| Recall | 0.7143 | 0.6429 |
| F1 | 0.7477 | 0.7129 |
| Error rate | 0.3333 | 0.3580 |

Table 20: Confusion Matrix for the Sensitivity analysis of the optimized Random Forest

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Predicted Values** | | | |
|  |  | **N of trees = 100** | | **Max depth = 5** | |
|  |  | **Negative** | **Positive** | **Negative** | **Positive** |
| **Actual Values** | Negative | TN:14 | FP:11 | TN:16 | FP:9 |
| Positive | FN: 16 | TP:40 | FN: 20 | TP:36 |

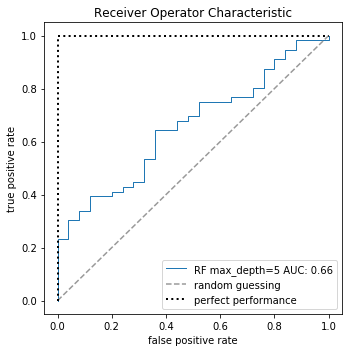
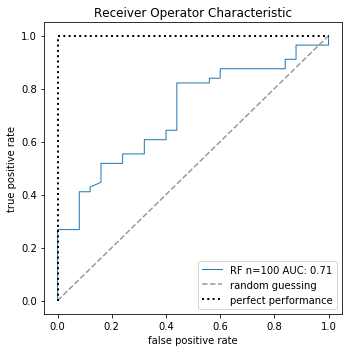


Figure 15: ROC-AUC at different number of trees and maximum depth and gamma for the optimized Random Forest.

**Discussion and Conclusion**

SVM results from Ramana *et al* (5) contrasts highly with the ones obtained in this work, reaching very high scores in every evaluation criteria used. As an example, their highest Accuracy score was of 97.86% for the first 5 features ordered by importance. This great difference against the results obtained here suggests that there could be issues with their data preprocessing like class rebalancing, feature scaling or the proportion of training and test data after splitting, since there is no mention of these steps in their work.

With respect to results obtained with RF algorithms, Pahareeya *et al* (9) reached high levels of accuracy by oversampling the minority class by 200%, resulting in a score of 89.12% in contrast with the 67.9% achieved in this work. Nevertheless, it seems that this oversampling was done before a data split, which could create data leakage.

In this work, a high number of false negatives was obtained with both optimized SVM and RF algorithms. This is not acceptable in a medical context, since these incorrectly classified patients would be at risk of being overlooked by medical professionals and develop a more advanced liver disease. To deal with this issue a number of alternatives arise. For example, the class imbalances could be dealt in a different manner, not trying to reach a 1:1 ratio between patients and non-patients with synthetic samples, but this could affect the evaluation since the algorithm shouldn’t be exposed to balanced testing data. An alternative to this is to undersampling the majority class, but since there are already a small number of instances to work with, this option could cause issues in the training. An option is to gather more data from non-patients to balance the dataset, and from this point generate a new classification model.

It has to be said that this dataset is relatively small, so the proportion of data between training and testing sets could affect the results since small test sets can create bias in results obtained. It’s possible that a 85%-15% split is not optimal, and other proportions would need to be explored.

Other classification algorithms could be optimized for the dataset, and it’s possible to create an ensemble of these optimized classifiers to obtain better results.

The potential for deployment of an algorithm to facilitate the diagnosis of liver disease could be of high impact in zones where access to medical professionals is limited, be it in developing countries or where demand for endocrinologists highly surpasses the delivering capabilities of health institutions. But as Ramana *et al* write (10) there are differences between populations that could be crucial, so it is possible that a trained algorithm that works with high accuracy in one population could struggle in another one. This could be because of intrinsic characteristics of the underlying population.

Two optimized classification algorithms were presented in this work, but they didn’t reached a level where is feasible to apply them in a real setting, and more alternatives need to be explored.

**References**

1. Liver function tests: MedlinePlus Medical Encyclopedia". www.nlm.nih.gov. Retrieved 7-5-2019.

2. Liver disease – NHS Choices. www.nhs.uk. Retrieved 7-5-2019.

3. Ramana. ILPD (Indian Liver Patient Dataset) Data Set. Department of Information Technology, Aditya Instutute of Technology and Management, Tekkali - 532201, Andhra Pradesh, India. http://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29#

4. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

5. Bendi Venkata Ramana, Prof. M. S. Prasad Babu and Prof. N. B. Venkateswarlu, A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis, International Journal of Database Management Systems (IJDMS), Vol.3, No.2, ISSN: 0975-5705, PP 101-114, May 2011.

6. N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, 321-357, 2002.

7. Cortes, C., Vapnik, V., Support-Vector Networks. Machine Learning, 20, 273-297 (1995).

8. Breiman, L., Random Forests. Machine Learning, 45, 5–32, (2001).

9. Pahareeya, J., Vohra R., Makhijani J., Patsariya S. Liver Patient Classification using Intelligence Techniques. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 4, Issue 2, February 2014.

10. Ramana, Prof. M. S. Prasad Babu and Prof. N. B. Venkateswarlu, A Critical Comparative Study of Liver Patients from USA and INDIA: An Exploratory Analysis, International Journal of Computer Science Issues, ISSN :1694-0784, May 2012.

11. Pedersen, R., & Schoeberl, M. (2006). An Embedded Support Vector Machine. In Proceedings of the Fourth Workshop on Intelligent Solutions in Embedded Systems (WISES 2006) (pp. 79-89)

12. Bernstein, M., Random Forests, Computer Science Department, University of Wisconsin. http://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/ensembles/RandomForests.pdf