Capstone Project

Predicting Liver disease from data

Machine Learning Engineer Nanodegree

KOTHAKOTA SRINATH

05-02-2019

Definition

Project Overview :

In India, delayed diagnosis of diseases is a fundamental problem due to a shortage of medical professionals. A typical scenario, prevalent mostly in rural and somewhat in urban areas is:

1.A patient going to a doctor with certain symptoms.

2.The doctor recommending certain tests like blood test, urine test etc depending on the symptoms.

3.The patient taking the aforementioned tests in an analysis lab.

4.The patient taking the reports back to the reports back to the hospital, where they are examined and the disease is identified.

The aim of this project is to somewhat reduce the time delay caused due to the unnecessary back and forth shuttling between the hospital and the pathology lab. Historically, work has been done in identifying the onset of diseases like heart disease, Parkinson's from various features, for example in this paper https://www.springer.com/gp/book/9781468443875 this case, a machine learning algorithm will be trained to predict a liver disease in patients.

Problem Statement

The problem statement is formally defined as:

'Given a dataset containing various attributes of 583 Indian patients, use the features available in the dataset and define a supervised classification algorithm which can identify whether a person is suffering from liver disease or not.'

The dataset for this problem is the ILPD (Indian Liver Patient Dataset) taken from the UCI Machine Learning Repository . Number of instances are 583. It is a multivariate data set, contain 10 variables that are age, gender, total Bilirubin, direct Bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT and Alkphos. All values are real integers. This data set contains 416 liver patient records and 167 non- liver patient records.The data set was collected from north east of Andhra Pradesh, India. This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age "90".

→Here for this type of problem I use the classification models to get the good results. I use the logistic,Randomforest,SVM and kNeighbours by using this I conclude that which will work better and finalize the model that will predict the good results for liver diseases prediction.

Relevant sources:

1.Bendi Venkata Ramana, Prof. M. S. Prasad Babu and Prof. N. B. Venkateswarlu, â€œA Critical Comparative Study of Liver Patients from USA and INDIA: An Exploratory Analysisâ€•, International Journal of Computer Science Issues, ISSN :1694-0784, May 2012.

Strategy

This seems to be a classic example of supervised learning. We have been provided with a fixed number of features for each data point, and our aim will be to train a variety of Supervised Learning algorithms on this data, so that , when a new data point arises, our best performing classifier can be used to categorize the data point as a positive example or negative. Exact details of the number and types of algorithms used for training is included in the 'Algorithms and Techniques' sub-section of the 'Analysis' part.

Metrics :

In problems of disease classification like this one, simply comparing the accuracy, that is, the ratio of correct predictions to total predictions is not enough. This is because depending on the context like severity of disease, sometimes it is more important that an algorithm does not wrongly predict a disease as a non-disease, while predicting a healthy person as diseased will attract a comparatively less severe penalty.

Thus, here we will use classification_report.In this we contain precision ,recall,f1-score,support and avg/total. Based on the this f-score and accuracy we find the good model because if we predict the healthy person as diseased it is not good.so,we must carefull in finalizing the model based on their performance ,accuracy,f-scores.

Thus, here we will use F-beta score as a performance metric, which is basically the weighted harmonic mean of precision and recall. Precision and Recall are defined as:

Precision=TP/ (TP+FP), Recall=TP/ (TP+FN), where

TP=True Positive

FP=False Positive

FN=False Negative

F-beta score is:

F-beta score = (1+β2)*precision*recall/((β 2*precision)+recall)

β = A number that decides relative weightage of precision and recall. In this case, a disease being classified as a non-disease will incur a high penalty. So, more emphasis is placed on recall.

Analysis

Exploring the Data

The ILPD dataset contains ten features as listed below:

1. Age

2. Gender

3. Total bilirubin

4. Direct bilirubin

5. Total proteins

6. Albumin

7. A/G ratio

8. SGPT

9. SGOT

10. Alkphos

All features, except Gender are real valued integers. The last column, Disease, is the label (with '1' representing presence of disease and '2' representing absence of disease). Total number of data points is 583, with 416 liver patient records and 167 non liver patient records.

In the description of the dataset, it is observed that some values are 'Null' for the 'Álkphos' column. Accordingly, 4 rows containing those values are removed and replaced with 0.

. A brief description of dataset, including parameters like mean, min, max for each column is given below:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Am
count	583	583	583	583	583	583	
unique	NaN	2	NaN	NaN	NaN	NaN	
top	NaN	Male	NaN	NaN	NaN	NaN	
freq	NaN	441	NaN	NaN	NaN	NaN	
mean	44.74614	NaN	3.298799	1.486106	290.5763		
std	16.18983	NaN	6.209522	2.808498	242.938		
min	4	NaN	0.4	0.1	63		
25%	33	NaN	0.8	0.2	175.5		
50%	45	NaN	1	0.3	208		
75%	58	NaN	2.6	1.3	298		
max	90	NaN	75	19.7	2110		

Dataset I have taken:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotra
0	65	Female	0.7	0.1	187	
1	62	Male	10.9	5.5	699	
2	62	Male	7.3	4.1	490	

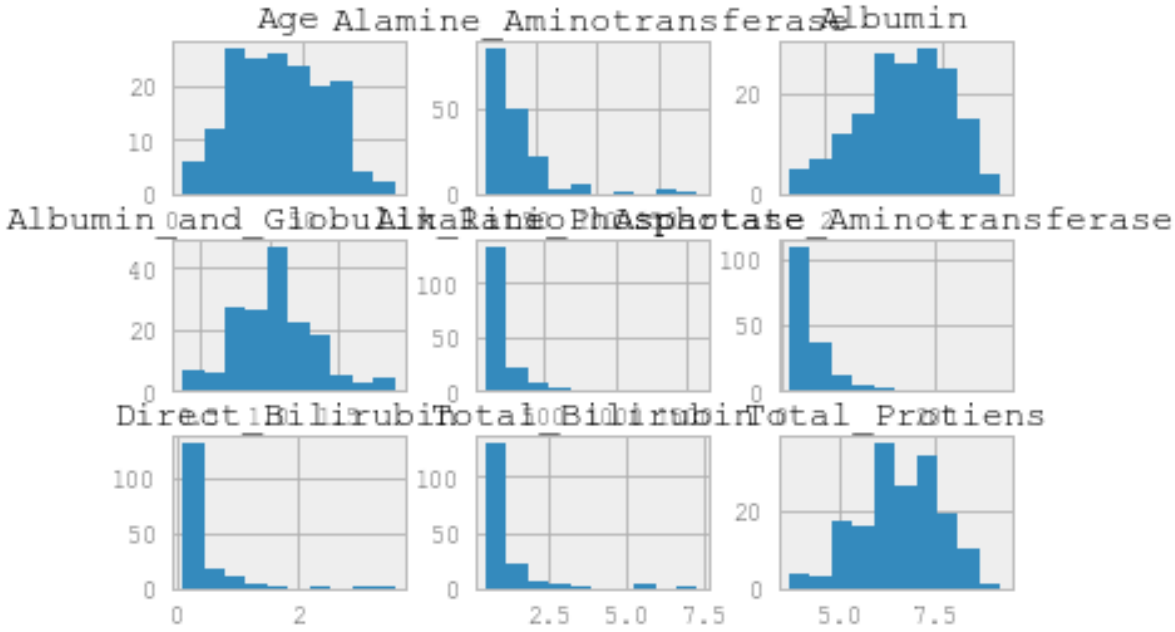
3	58	Male	1	0.4	182	
4	72	Male	3.9	2	195	
5	46	Male	1.8	0.7	208	
6	26	Female	0.9	0.2	154	
7	29	Female	0.9	0.3	202	
8	17	Male	0.9	0.3	202	
9	55	Male	0.7	0.2	290	
10	57	Male	0.6	0.1	210	
11	72	Male	2.7	1.3	260	
12	64	Male	0.9	0.3	310	
13	74	Female	1.1	0.4	214	
14	61	Male	0.7	0.2	145	
15	25	Male	0.6	0.1	183	
16	38	Male	1.8	0.8	342	
17	33	Male	1.6	0.5	165	
18	40	Female	0.9	0.3	293	
19	40	Female	0.9	0.3	293	
20	51	Male	2.2	1	610	
21	51	Male	2.9	1.3	482	
22	62	Male	6.8	3	542	
23	40	Male	1.9	1	231	
24	63	Male	0.9	0.2	194	
25	34	Male	4.1	2	289	
26	34	Male	4.1	2	289	
27	34	Male	6.2	3	240	
28	20	Male	1.1	0.5	128	
29	84	Female	0.7	0.2	188	
...	...	...	...	...	...	...
553	46	Male	10.2	4.2	232	
554	73	Male	1.8	0.9	220	
555	55	Male	0.8	0.2	290	
556	51	Male	0.7	0.1	180	
557	51	Male	2.9	1.2	189	
558	51	Male	4	2.5	275	
559	26	Male	42.8	19.7	390	
560	66	Male	15.2	7.7	356	
561	66	Male	16.6	7.6	315	
562	66	Male	17.3	8.5	388	
563	64	Male	1.4	0.5	298	
564	38	Female	0.6	0.1	165	
565	43	Male	22.5	11.8	143	
566	50	Female	1	0.3	191	
567	52	Male	2.7	1.4	251	
568	20	Female	16.7	8.4	200	
569	16	Male	7.7	4.1	268	

570	16	Male	2.6	1.2	236	
571	90	Male	1.1	0.3	215	
572	32	Male	15.6	9.5	134	
573	32	Male	3.7	1.6	612	
574	32	Male	12.1	6	515	
575	32	Male	25	13.7	560	
576	32	Male	15	8.2	289	
577	32	Male	12.7	8.4	190	
578	60	Male	0.5	0.1	500	
579	40	Male	0.6	0.1	98	
580	52	Male	0.8	0.2	245	
581	31	Male	1.3	0.5	184	
582	38	Male	1	0.3	216	

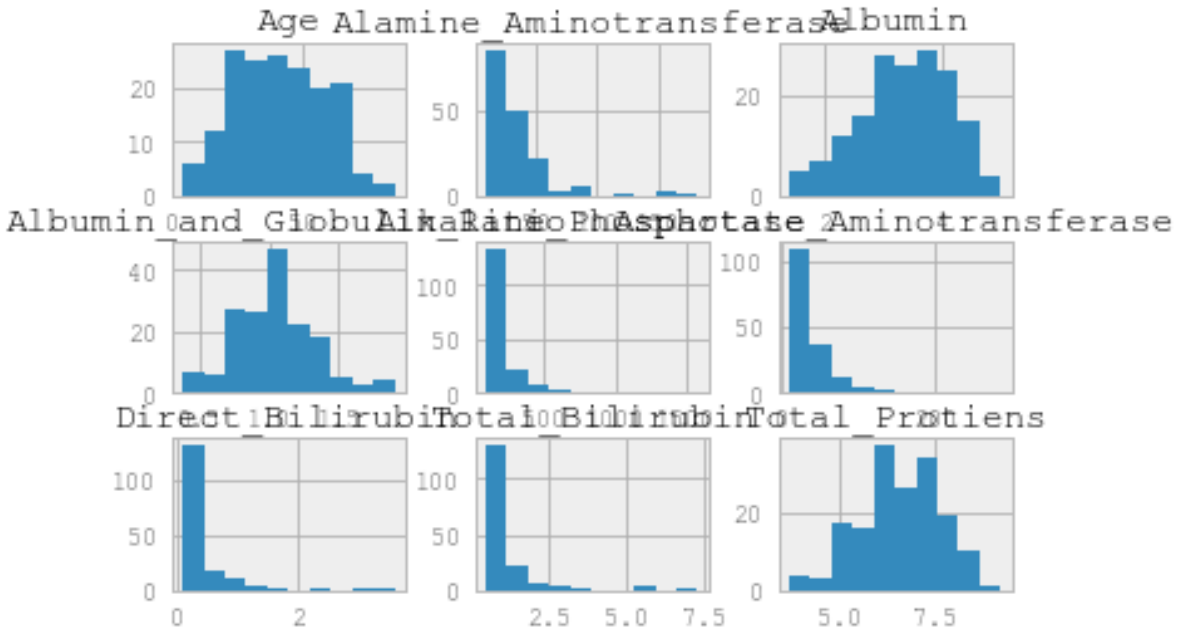
You can also see the dataset in this link: https://www.kaggle.com/jeevannagaraj/indian-liver-patient-dataset

Exploratory Visualization

After removing the column 'Dataset' from the dataset as it is the label, we display all features in a histogram format to check if any feature is skewed (contains a small number of outlier values).



Skewed features found are Aspartate_Aminotransferase, Alkaline_Phosphotase,Alamine_Aminotransferase, Total Bilirubin, Direct_Bilirubin . On these, a log transformation is applied to reduce their range. Again, all the transformed features are shown in a histogram format:



Algorithms and Techniques

Three supervised learning approaches are selected for this problem. Care is taken that all these approaches are fundamentally different from each other, so that we can cover as wide an umbrella as possible in term of possible approaches. For example- We will not select Random Forest and Ada Boost together as they come from the same family of 'ensemble' approaches. The choice of algorithms was influenced from these two sources:

https://stackoverflow.com/questions/2595176/which-machine-learning-classifier-to-choose-in-general

For each algorithm, we will try out different values of a few hyperparameters to arrive at the best possible classifier. This will be carried out with the help of grid search cross validation technique. The algorithms are described below:

1.Random Forest Classifier:

It comes under the category of ensemble methods. It employs 'bagging' and 'boosting' methods to draw a random subset from the data, and train a Decision Tree on that (hence, the name Random Forest). In this case, we don't know the relative importance of each feature while deciding the output, so a Random Forest can be successful as it will ensure training on different randomized subsets. Hyperparameters to be manipulated:

 n_estimators(number of trees in a forest) max_depth(maximum depth of one single tree) criterion.

Pros:

 Through combining base learners, Random forest improves the performance of the weak algorithm. Random forests are important when there are multiple correlated features.

Cons:

 Takes greater computational time to train. Has a tendency to overfit especially if number of examples is less, unless hyperparametersare properly adjusted.

2.Support Vector Machine:

SVM aims to find an optimal hyperplane that separates the data into different classes, using a method called as kernel to project data points belonging to a particular class into different dimensions, so that a hyperplane can easily pass through and maintain the largest possible distance between itself and these data points. Hyperparameters to be manipulated:

 kernel(type of kernel used like 'linear', 'rbf' etc for separating data points) C(the penalty assigned to the error term)

Pros:

 Performs well with high dimensional data Performs well with non-linear boundary if appropriate kernel used

Cons:

 Expensive to train

3.K- Nearest Neighbors:

KNN classifies a given data point by looking at its neighbors and assigning weights to them in such a way that the closest neighbours have a greater say in determining the class. Here, distance can be Euclidean Distance, Minkowski Distance etc.

Hyperparameters to be manipulated:

 n_neighbors(number of neighbors) weights(the degree of influence various data points possess)

Pros:

 Simple to implement Flexible with regards to distance of data points

Cons:

 All heavy computational work takes place during testing

Benchmark

Some models have been created to analyze the chances of liver injury in critically ill patients. The following paper gives one such model: https://www.ncbi.nlm.nih.gov/pubmed/26820880. It assigns a LiFe score from 0 to 10 to patients: with 0 denoting low risk and >8 denoting very high risk. The authors of this paper have stated that: 'a significant positive correlation exists between LiFe score and acute-on-chronic liver failure grade, ($r = 0.478$, $P < 0.001$)'.

However the problem lies in finding a dataset where the results are given in such a fashion which is easily comparable with our classification values. In datasets like the one mentioned above, it is intrinsically difficult to compare the scores given with our outputs. Therefore, we will use a simple algorithm like Logistic Regression as our benchmark model and try to improve upon its performance by using other algorithms like SVM, ensemble methods etc.

Logistic Regression:

Since the outcome is binary and we have a reasonable number of examples at our disposal compared to number of features, this approach seems suitable. At the core of this method is a logistic or sigmoid function that quantifies the difference between each prediction and its corresponding true value. When presented with a number of inputs , it assigns different weights to features(based on their relative importance).Since for this data it already knows the output beforehand, it continuously adjusts the weights such that when these weights summed up with their features are introduced in the logistic function, the results are as near as possible to the actual ones. Once presented with a test value, it again inserts the value into our logistic function and returns the output as a number between 0 and 1, which represents the probability of that test value being in a particular class.

Beating this benchmark model means that our method is suitable to be applied in the real world, as the problem dataset inherently favours Logistic Regression in terms of limited sample size and large number of positive examples (people having the disease). In real world, a much greater dataset size can be created due to the large population, and the percentage of positive cases will also be quite less. In such cases, the algorithms chosen are known to perform better, and our assumption of placing more emphasis on recall will also be better placed. So, if any one of these models manages to beat or even have comparable performance metrics to Logistic regression, it will have a high probability of giving a better performance in a real world scenario.

Methodology

Data Preprocessing

As explained in the section 'Exploring the data', rows having 'Null' values were removed from the dataset and repalced with 0. Thereafter, log transformation was applied to features which were showing a skewed pattern

Thereafter, all columns in the dataset except ‘Gender’ are normalized. Transformation is given by:

df_norms=(df-df.mean())/(df.max()-df.min())

Then we use pd.get_dummies(df) method to one-hot encode the feature ‘Gender’ as well as the label ‘Disease’ (with the integer ‘1’ representing presence of disease and 2 represent the not presence of diseases).

After normalizing the data we implement the model because if we don’t remove the null it will effect the accuracy so,I removed them and replaced with 0 because our data set is small if we remove them it is not good to train and test the model.then we divide the total dataset into training and testing data sets randomly .after that we will fit the model and test the model to finalize that it will predict correct or not.

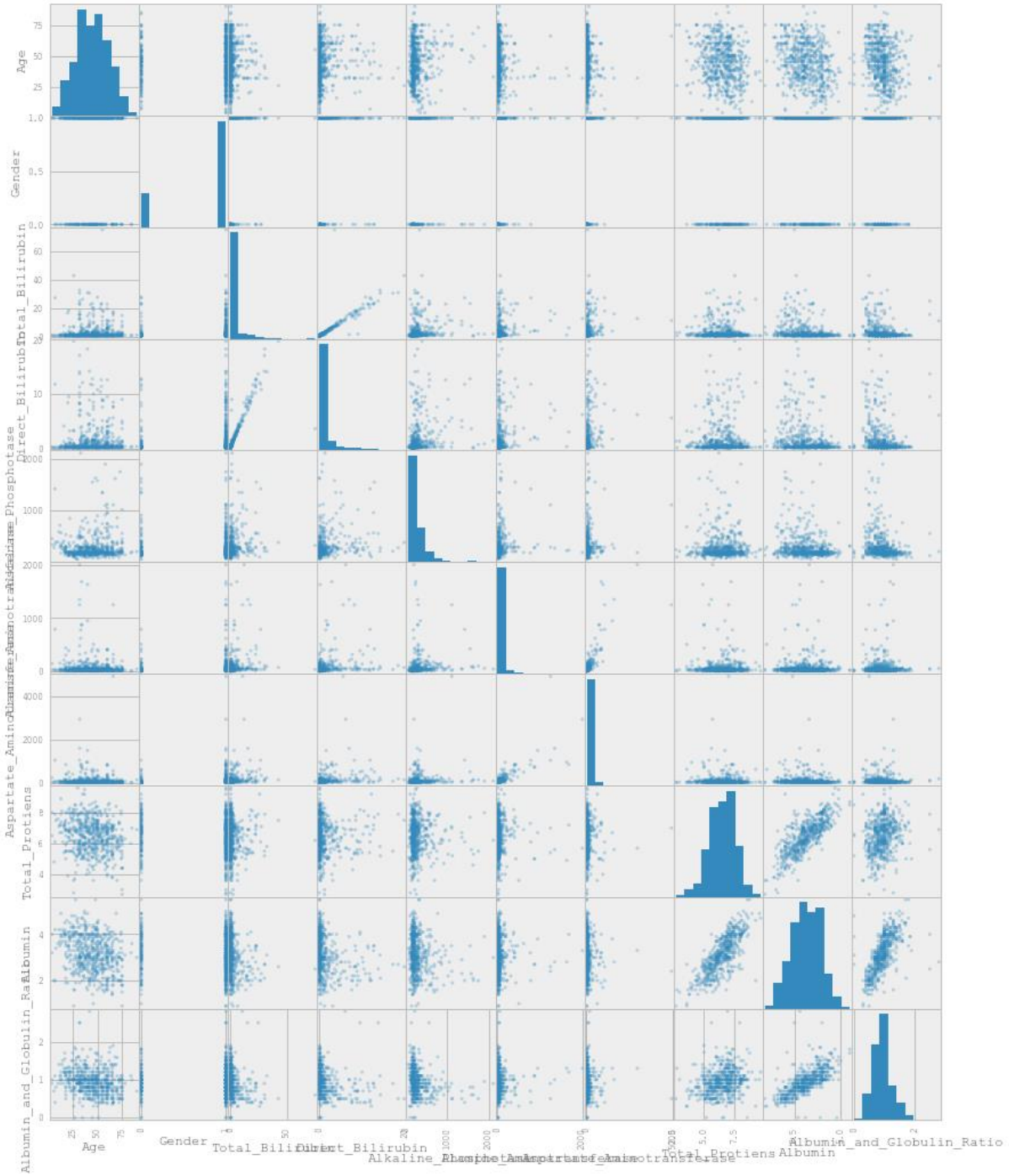
Implementation

The dataset will be split into training and testing set .the split using train_test_split method from sklearn. Random state will be specified as a particular number so that we have a means for comparison later, by specifying the same random state.

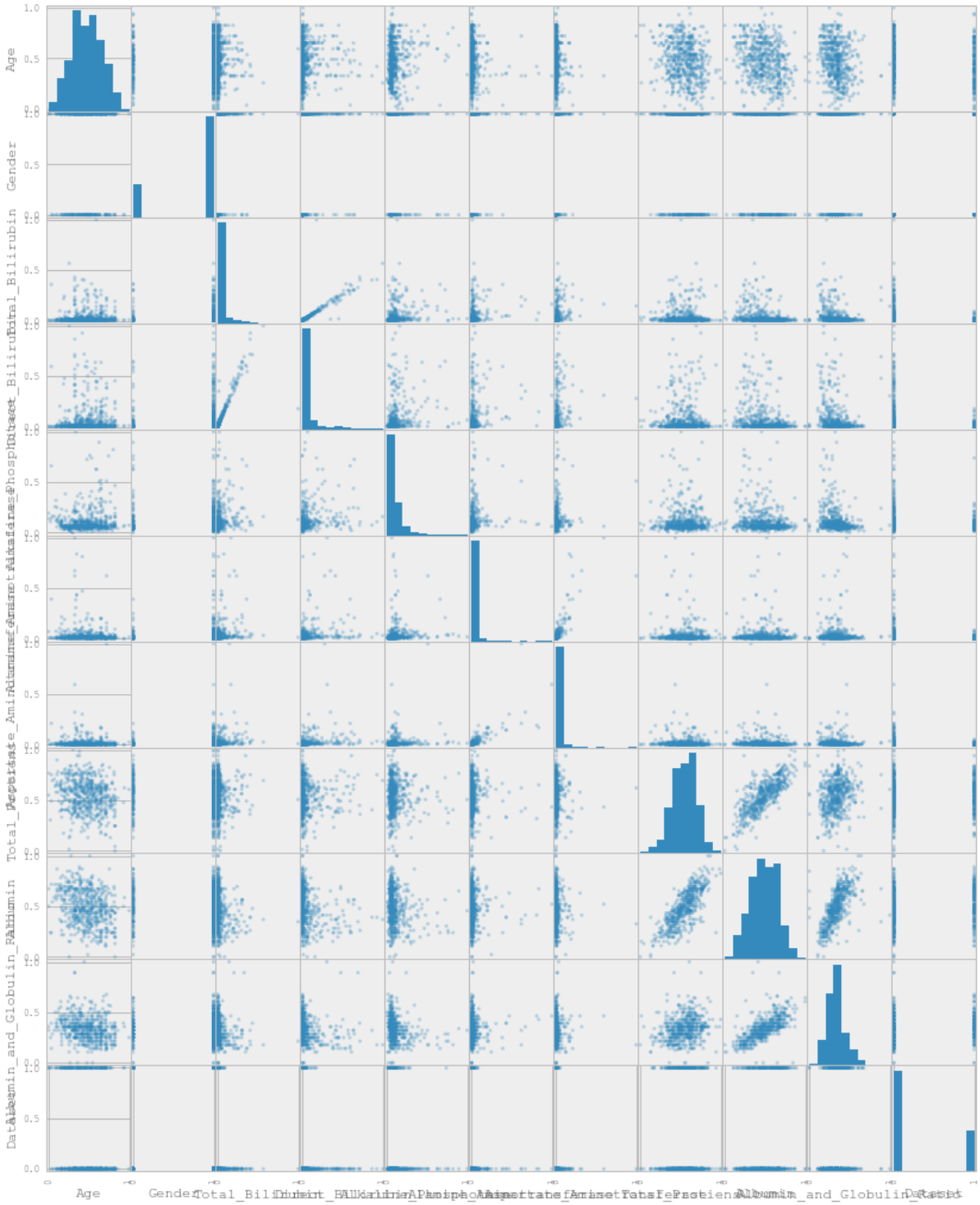
Before applying any supervised learning technique, we will implement a naïve predictor, that will simply return that every data point has ‘Disease’= True. We will check our accuracy on that predictor. Note that in this case naive predictor will perform artificially well unlike in real world, a large proportion of patients (around 70%) do have the disease.

Then, a method called as 'train_predict' is defined that takes as input the following: learner, sample_size, X_train, y_train, X_test, y_test. It returns the accuracy and F-beta score on training and testing set respectively.

Before normalizing:


Age
Gender
Total_Bilirubin
Direct_Bilirubin
Alkaline_Phosphotase
Aspartate_Aminotransferase
Total_Protiens
Albumin
Albumin_and_Globulin_Ratio

After normalizing:



Refining The Models

It is not very clear which model is the best performer: SVM gives the best scores on testing set, while Random Forest Classifier gives the less accuracy. So, we will test both these models on a variety of parameters using the GridSearchCV technique. For the sake of comparison, we will include KNN in the fray too. The number of parameters used is constrained by the computational time taken to compute the results. Hyperparameters and their values for different classifiers are:

Random Forest Classifier :

criterion:['gini','entropy'],n_estimators:[100,10]

estimator=model,param_grid=params,scoring='accuracy',cv=10

SVM :

kernel='linear',C=1

K nearest neighbors :

n_neighbors=4,weights='uniform'

Results

Model Evaluation and Validation

Since the size of dataset is small at present , there is not much difference between training and testing times of different algorithms. However, for the sake of comparison, these times have been displayed in the 'Implementation' sub-heading of 'Analysis' section.

Details of performance metrics for each classifier after implementing GridSearchCV and choosing the best combination of their parameter values is given below:

The support vector machine performs classification by finding the hyperplane that maximizes the margin between the two classes. I'm sure that my model will predict the unseen data, Based on generalization error this value tells about how a model can predict the values of unseen data. It is sensitive to change the outliers in this dataset, but one can trust this model because this model will predict correctly if a person have the disease or not .The SVM allows very low error in classification.

It dependes on selection of kernel,kernel parameters,soft margin parameter C to give the effectiveness. Based on this all we can trust the model.

RandomForest:

Accuracy:0.684931506849

	precision	recall	f1-score	support
1	0.96	0.93	0.95	86
2	0.82	0.90	0.86	31
avg / total	0.93	0.92	0.92	117

KNeighborsClassifier:

Accuracy:0.691428571429

	precision	recall	f1-score	support
1	0.95	0.73	0.83	108
2	0.15	0.56	0.23	9
avg / total	0.89	0.72	0.78	117

SVM:

Accuracy:0.709401709402

	precision	recall	f1-score	support
1	1.00	0.71	0.83	117
2	0.00	0.00	0.00	0
avg / total	1.00	0.71	0.83	117

conclusion:

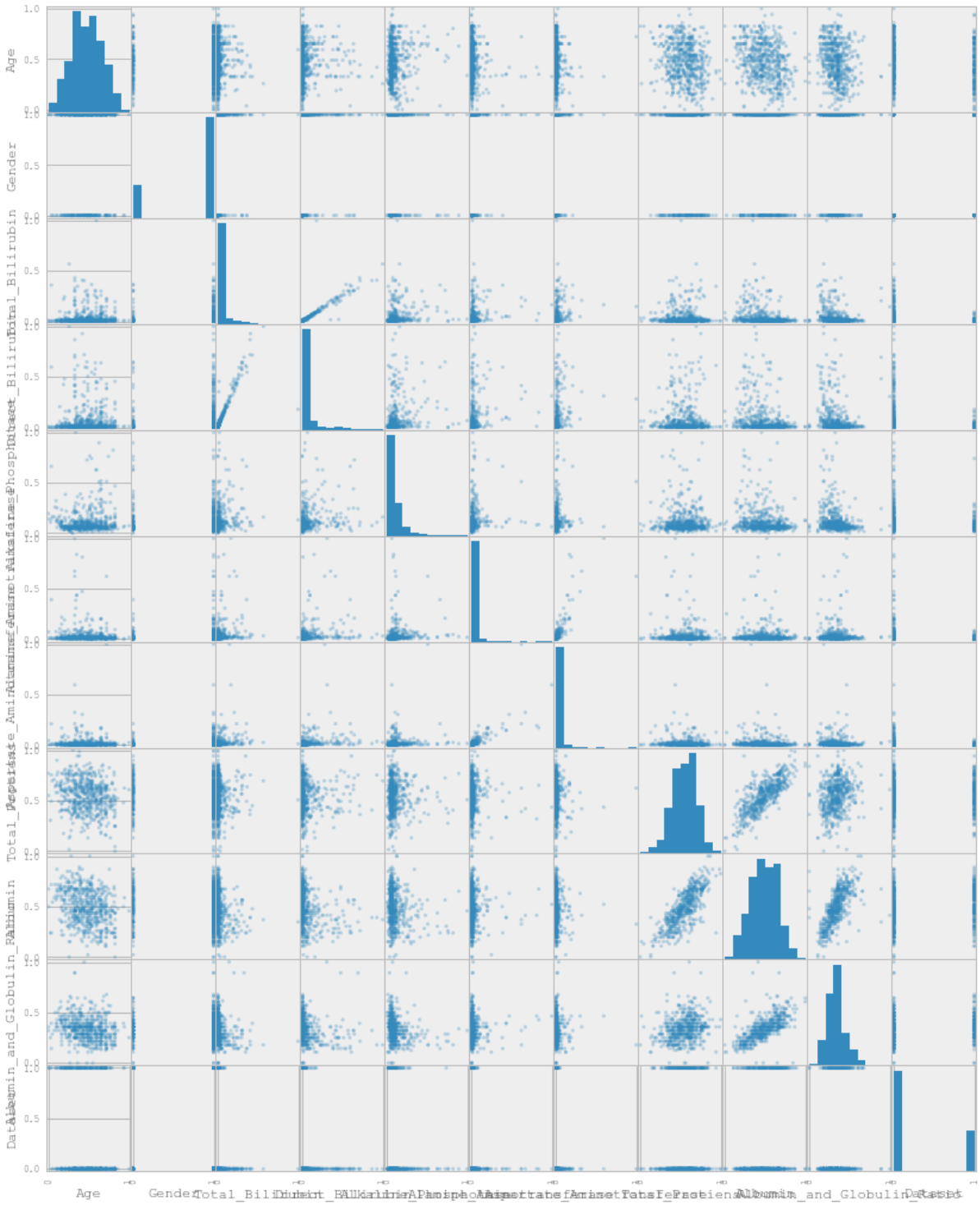
Based on the f-score and accuracy we conclude that the SVM is good model to predict the liver disease.

The nomalizing the data set will give the overview of data .

When I'm searching for dataset I really wonder that using this models which I learned in machine learning will help to predict the diseases.After seeing this I choose the liver diseases prediction

Because in india many poor people are suffering from liver diseases they don't have much many to cure the diseases if they go doctor it will take much time to predict the diseases. By using my model it will predict the disease in shorter time. It will save the time .for predicting the liver disease I used the models such as Random forest ,SVM, logistic and Kneighbours to predict the diseases out of this model I will choose the model that which will give more accuracy on predicting the diseases.

- ➔ By changing some features in my model it will helpful to get more accuracy and easy to predict. I recommend that use the large data set .It will helpful to divide the data set into train and test dataset this will give the more prediction.
- ➔ Some changes will improve my model . there are lot of things to do in SVM. For the performance analysis you can do cross validation(k-fold),and if performance is not adequate just change the kernel and try again.RBF kernel is agood choice .I can only suggest that R package 'caret' which has a very good tool to find the optimal hyperparameters for SVM.
- ➔ Remove the outliers and remove the null datapoints. Use the another parameters in the model to get more accuracy.



Justification

Despite the ambiguous results, it was decided to select SVM as the final classifier.even though the f-score of random forest

have more(0.95).the Svm predict the diseases with more accuracy then other models .

Having said that, only one model (SVM) managed to surpass our benchmark classifier of Logistic Regression, with Random Forest

coming close. As discussed before, this is probably due to the small size of the dataset and the very high number of positive

examples. Once the size of the dataset increases beyond a limit, the algorithm selected by us should be able to surpass Logistic

Regression.

From the results I conclude that the SVM model will work as efficient as remaining models.