**Acute Liver Failure Classifier**

Table of Contents

[Introduction 1](#_Toc120765041)

[Problem Statement 1](#_Toc120765042)

[Existing Approaches 1](#_Toc120765043)

[Similarities and Differences with Existing Work 2](#_Toc120765044)

[Importing the Data 3](#_Toc120765045)

[Preprocessing 3](#_Toc120765046)

[Analysis 4](#_Toc120765047)

[Applying Algorithms 9](#_Toc120765048)

[Model 1 – XGBoost 9](#_Toc120765049)

[Model 2 – Grid Search 10](#_Toc120765050)

[Model 3 – Decision Tree 10](#_Toc120765051)

[Evaluation of Algorithms 11](#_Toc120765052)

[Selection of Metrics 11](#_Toc120765053)

[Comparison of Algorithms Based on the Metrics 11](#_Toc120765054)

[Conclusion 13](#_Toc120765055)

[Bibliography 14](#_Toc120765056)

# Table of Figures

[Figure 1: Raw data had 30 columns and 8785 rows 3](#_Toc120765097)

[Figure 2: The raw data had columns of object data type 3](#_Toc120765098)

[Figure 3: Many columns had missing values 4](#_Toc120765099)

[Figure 4: Correlation matrix of variables 5](#_Toc120765100)

[Figure 5: Correlations of variable to ALF 6](#_Toc120765101)

[Figure 6: Visualizing distribution of data 7](#_Toc120765102)

[Figure 7: Pairplot of the features 8](#_Toc120765103)

[Figure 8: Skew in ALF results 9](#_Toc120765104)

[Figure 9: Confusion matrices of the algorithms 12](#_Toc120765105)

# Introduction

## Problem Statement

Acute liver failure (ALF) is a sudden and rare illness that occurs in patients who previously didn’t have existing liver disease (Bernal & Wendon, 2013). ALF mostly occurs in adults over the age of 30. The disease is much rare in the developed world compared to developing nations, where viruses such as hepatitis A and E have been attributed to more than 50% of the deaths of individuals from the illness (Hoofnagle, Nelson, & Purcekk, 2012). The diseases possess a high fertility rate of between 65% to 85% (Batra & Acharya, 2003). With advances in liver transplant technology and intensive care, the fatalities possed by the disease have been reduced. This, however, is a costly treatment.

Prediction of ALF can be useful for patients to take corrective action in the face of the illness. Also, caregiving institutions can benefit from this prediction in enabling them to mobilize the resources needed to mitigate the illness in time.

This assessment aims at building a classifier using machine learning algorithms to help in the prediction of acute liver failure.

## Existing Approaches

Little work has been done on the use of machine learning in the prediction and diagnosis of acute liver failure. The existing methods used to diagnose acute liver failure are:

* Blood tests – a prothrombin test is used to determine how well the liver works. A long time to clot indicates that the liver is not functioning well.
* Imaging tests – ultrasound, CT scans, or MRI scans can be used to check for markers that indicated liver failures such as tumors or scarring.
* Biopsy – a sample of the liver is obtained to be tested. Extra care has to be taken since blood clotting is difficult for people with ALF.

## Similarities and Differences with Existing Work

ALF is a rare condition and as such machine learning has not been well studied for the prediction and diagnosis of the illness. A work was found in which machine learning was examined for the detection of ALF (Rafi, 2022). In his work, a classifier was built to predict ALF using machine learning algorithms. XGBoost and Random Forest were compared as machine learning algorithms. Accuracy was used as the main metric in his work, and the Random Forest Tree was found to have the highest accuracy of 92.1333.

The first weakness that has discovered in the work was in feature selection. A sample of features was selected seemingly randomly. In this work, Pearson correlation is used to help in the selection of features. This way, the model produced is to have a better performance.

Also, the skew in data was not considered in their work. It was found that the results of ALF were skewed towards the negative – more people in the data set tested negative for ALF. To mitigate this, random oversampling was used on the training data in this work.

An important weakness that was discovered in the work was that accuracy was used as the metric for evaluating the algorithms. In such a case where data is skewed, and where the consequences of false negatives are high, recall serves as a better metric. In this work, recall is the main metric to consider. Also, the F1-Score is considered, although with moderate importance to the evaluation.

Finally, this work considered three algorithms: XGBoost, Random Forest, and Decision Tree.

# Importing the Data

First, the relevant modules were imported into the notebook. The data set, which was in *.xlsx* format, was imported as a dataframe. The raw dataset had 30 columns and 8785 rows.



Figure : Raw data had 30 columns and 8785 rows

# Preprocessing

A summary of the columns of the dataframe was obtained using the *df.info()* function. It was found that some columns were of *object* type as shown in figure 2.

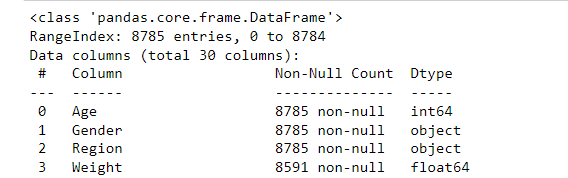


Figure : The raw data had columns of object data type

These columns in reality held *string* data. An array containing the identifiers of these columns was obtained and converted to columns of type *string*.

cols\_to\_convert = df.select\_dtypes(include="object").columns

df[cols\_to\_convert] = df[cols\_to\_convert].astype("string")

The *Gender* column had two values: ‘M’ for male and ‘F’ for female. These values were converted to 0 and 1 respectively.

df.Gender[df.Gender == 'M'] = **0**

df.Gender[df.Gender == 'F'] = **1**

df['Gender'] = pd.to\_numeric(df['Gender'])

Missing values were searched for. It was found that a number of columns had missing values as shown in figure 3. Outright, the rows in which the target variable (*ALF*) was missing were dropped using:

df = df.dropna(axis = **0**, subset=['ALF'])

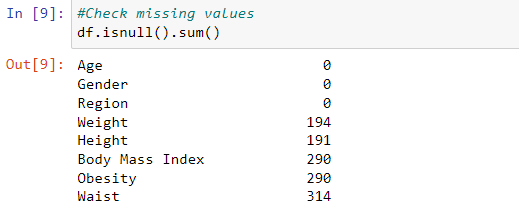


Figure : Many columns had missing values

Some columns, through intuition, were outright not relevant to the research problem. These columns were dropped too.

An array of the columns which had missing values were obtained and using *SimpleImputer*, filled to the median of the corresponding columns. Median was preferred as a strategy to avoid skewing of the data.

imputer = SimpleImputer(missing\_values=np.nan, strategy="median")

cols\_to\_handle = [**2**,**3**,**4**,**5**,**6**,**7**,**8**,**9**,**10**,**11**,**14**,**15**,**17**,**19**,**21**,**22**,**23**,**24**]

imputer = imputer.fit(df.iloc[:,cols\_to\_handle])

df.iloc[:,cols\_to\_handle] = imputer.transform(df.iloc[:,cols\_to\_handle])

# Analysis

A correlation matrix of the variables remaining in the cleaned table was generated and visualized in a heatmap as shown in figure 4. A striking correlation of other variables with the target variable – *ALF* – couldn’t be visualized from the heatmap. A few variables such as age seemed to have a reasonable degree of correlation. The heatmap was, however, a bit too cramped.

From the correlation matrix, the correlations of the variables to only *ALF* were then extracted as shown in figure 5. Of the variables, it was found that Age, with a correlation of 0.367639, had the highest positive correlation to ALF. Physical activity had the highest negative correlation with a value of -0.119604.

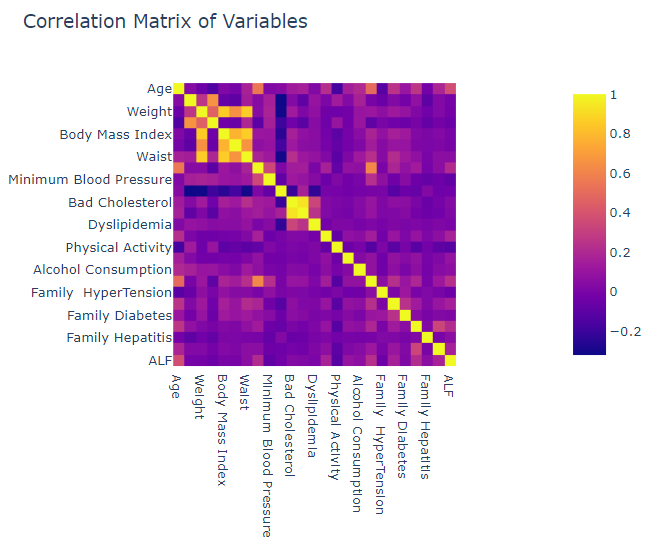


Figure : Correlation matrix of variables

The variables having a correlation of more than 0.2 or less than 0.2 were chosen as input features. For this assessment, the features chosen were: Age, HyperTension, Hepatitis, and Maximum Blood Pressure.

The pairplot of the features is shown in figure 7.

It was necessary to discover if there was any skew in the data. Plots of were generated for the distribution of the data as shown in figure 6. A striking realization was that there existed a skew in the ALF data. A count of the values of the ALF data was done, and it was found that 5536 values were negative results (0.0) while 464 values were positive (1.0) as shown in figure 8.. Moving onto the modeling phase of the assessment, this skew had to be kept in consideration.

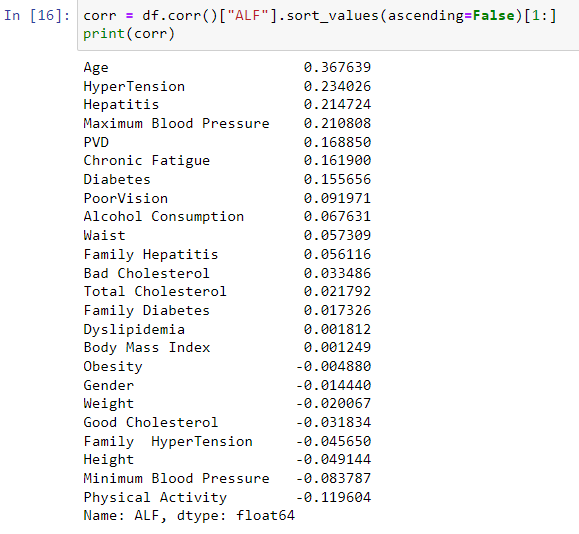


Figure : Correlations of variable to ALF



Figure : Visualizing distribution of data



Figure : Pairplot of the features

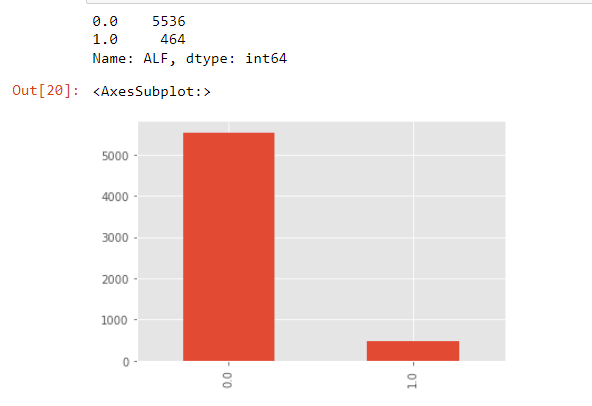


Figure : Skew in ALF results

# Applying Algorithms

The working data set was first split into training and testing components, with the training set being 0.25 of the overall data.

X\_train, x\_test, y\_train , y\_test = train\_test\_split(X,y, test\_size=**0.25** , random\_state=**42**)

To take care of the skew in ALF data that was observed in the previous phase, the training data was oversampled in using the minority strategy.

oversample = RandomOverSampler(sampling\_strategy='minority')

X\_balanced, y\_balanced = oversample.fit\_resample(X\_train, y\_train)

## Model 1 – XGBoost

XGBoost, with a learning rate of 0.1 and a random state of 10 was trained with the training set.

#XGBoost

xg\_cl = XGBClassifier(eval\_metric= 'error', learning\_rate= **0.1**, random\_state=**10**)

xg\_cl.fit(X\_balanced, y\_balanced)

#Prediction with training and testing data sets

xg\_pred\_train = xg\_cl.predict(X\_balanced)

xg\_pred\_test = xg\_cl.predict(x\_test)

## Model 2 – Grid Search

A Grid Search model was trained with the training set.

#Grid Search

rf = RandomForestClassifier(random\_state=**10**)

rf.fit(X\_balanced,y\_balanced)

gscv = GridSearchCV(estimator=rf, param\_grid=param\_grid, cv= **2**)

gscv.fit(X\_balanced, y\_balanced)

#Prediction with training and test data sets

gscv\_pred\_train = gscv.predict(X\_balanced)

gscv\_pred\_test = gscv.predict(x\_test)

Where the grid parameters were:

param\_grid = {

'n\_estimators': [**20**, **60**],

'max\_features': ['auto', 'sqrt'],

'max\_depth' : [**2**,**4**,],

'criterion' :['gini', 'entropy']

}

## Model 3 – Decision Tree

A decision tree with a maximum of 4 features and a maximum depth of 5 was trained with the training set.

#Decision Tree

dt =DecisionTreeClassifier(max\_features=**4** , max\_depth=**5**,criterion = 'entropy', random\_state=**0**)

dt.fit(X\_balanced, y\_balanced)

#Prediction with training and test data sets

dt\_pred\_train = dt.predict(X\_balanced)

dt\_pred\_test = dt.predict(x\_test)

# Evaluation of Algorithms

## Selection of Metrics

The objective of this classifier is to predict whether an individual is at risk of having acute live failure based on some features. ALF is sudden and has a high mortality rate. This means that the repercussions of failing to detect this are high. False negatives have a grim consequence in this setting. As such, the **recall** of the classifier is the most important metric for this classifier.

The importance of recall over accuracy in this setting is because the results are usually skewed in favor of the negative. Most patients don’t have ALF. A high accuracy, therefore, doesn’t necessarily mean that the classifier is good at detecting what it’s meant to – ALF.

The cost of false positives is not as high in this setting. Assuming that a patient is wrongfully diagnosed as having ALF by the classifier, it is expected that further tests are to be conducted to ascertain this. As such, the **f1-score** is a moderately important metric for this classifier.

The **confusion matrix** is can be used to visualize the metrics of the classifier.

### Comparison of Algorithms Based on the Metrics

A comparison of XGBClassifier, Grid Search, and Decision Tree on the basis of recall and f1-score metrics is shown in table 1 below.

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Recall** | **F1-Score** |
| **XGBClassifier** | 0.717 | 0.423 |
| **Grid Search** | 0.833 | 0.440 |
| **Decision Tree** | 0.819 | 0.450 |

Table : Performance of classifiers based on recall and f1-score

The confusion matrices for the classifiers are shown in figure 9.

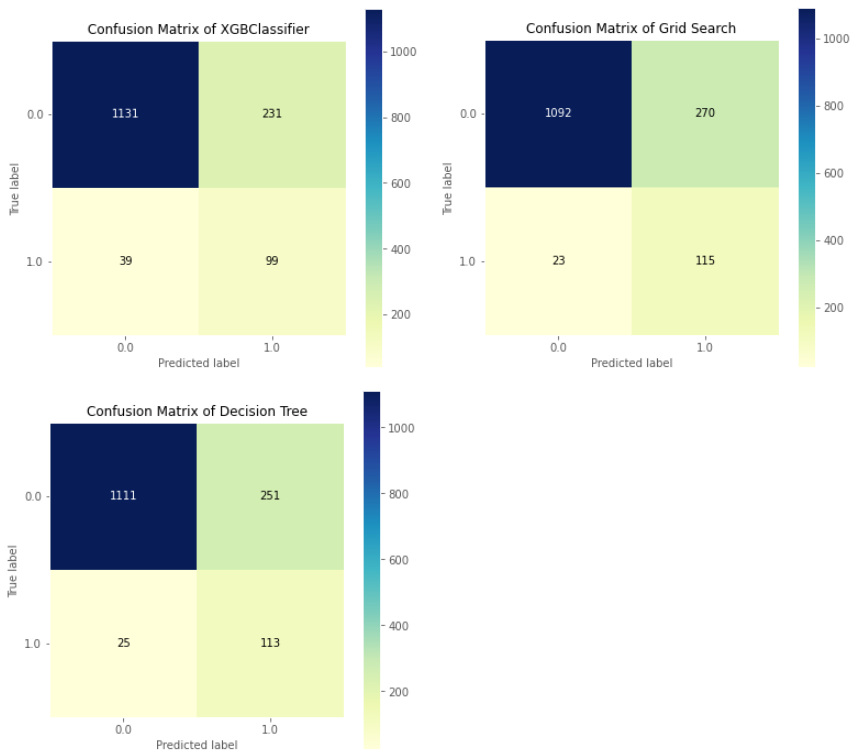


Figure : Confusion matrices of the algorithms

Having rationalized why the recall is the most important metric in this setting, it can be concluded that of the three algorithms, grid search, with a recall of 0.833, is the best algorithm for the classifier.

# Conclusion

The purpose of this assessment was to build a classifier to predict acute liver failure (ALF). The raw data imported into the Jupyter environment was cleaned to remove rows where ALF was missing. The missing values in the data set were filled using the median strategy to avoid skewing the data. Feature selection was accomplished using the Pearson correlation. The feature chosen had an absolute correlation of greater than two. The data was found to be skewed with the ALF results having more negative cases than positive ones. To handle this, random oversampling was done on the minority class. XGBoost, Search Grid, and Random Forest were investigated as machine learning algorithms for the classifier. Recall was chosen to be the most important metric due to the skew in the data and to since false negatives were seen to have a high consequence in the setting. It was found that Search Grid offered the best solution with a recall of 0.833.

|  |
| --- |
| **Bibliography** |

# Bibliography

Batra, Y., & Acharya, S. (2003). Acute liver failure: prognostic markers. *Indian J Gastroenterol*.

Bernal, W., & Wendon, J. (2013). Acute Liver Failure. *The New England Journal of Medicine*, 2525-2534.

Hoofnagle, J., Nelson, K., & Purcekk, R. (2012). Hepatitis E. *The New England Journal of Medicine*, 1237-1244.

Rafi, S. (2022). *Acute Liver Failure Prediction 2019 BUCC Workshop*. Retrieved from kaggle.