Survival Prediction for Patients with HCC

Julia Lynn
Brown University
December 9, 2022
https://github.com/Juliaclynn/HCC_Survival_Prediction.git

Introduction

Doctors have access to a lot of data about their patients – demographics, symptoms, lab results, etc. However, it is difficult and time-consuming for a doctor to synthesize all of that data when deciding on a treatment plan for a patient; the goal of this project was to design a model that could do some of that analysis for them, and transform raw clinical measurements into a more meaningful metric.

I designed a classification model to predict the survival of a patient one year in the future, using data collected at the Coimbra Hospital and University Centre in Portugal from 165 patients with hepatocellular carcinoma (HCC), the most common form of liver cancer. According to the World Health Organization, liver cancer was the third most common cause of cancer death in 2020 (WHO 2022). Any tool we can provide to assist doctors in their decision-making when treating this disease has the potential to help thousands of patients.

This data set is small (165 data points), which is typical for medical data. It has 49 features, corresponding to information available to and commonly used by doctors when evaluating HCC patients: 23 categorical features, 3 ordinal features, and 33 continuous features. (For a list of features, please refer to the description file in the 'data' folder.) The data set has a significant amount of missing data; less than 5% of data points contain data for all features, and 90% of features are missing data.

This data was originally used in a 2015 study published in the *Journal of Biomedical Informatics* which investigated new methods for neural network and logistic regression models with data sets containing missing data (Santos et al. 2015). Using cluster-based oversampling the authors were able to improve on existing models, with mean accuracy, AUC, and F1 values of 0.752, 0.700, and 0.665, respectively for a neural network model, and .730, .673, and .652, respectively for a logistic regression model. In a 2019 study published in *Cognitive Systems Research*, this data was used to test a new algorithm for feature selection and parameter optimization in a model to detect the presence of HCC (Książek et al. 2019). The authors found that their approach did improve accuracy, with accuracy and F1 score values of 0.885 and 0.876, respectively. In both studies, the authors imputed missing values (using nearest neighbor and average/modal imputation); with this project, I investigated model performance using methods other than imputation to handle missing values, specifically reduced-feature modeling and XGBoost models.

Exploratory Data Analysis

The distribution of the target variable in this data set is slightly imbalanced, with 102 data points (representing 61.8% of the data set) in class 1 (alive at one year), and 63 data points in class 0 (deceased before one year).

As mentioned above, the features in this data set are measurements commonly used by doctors to evaluate cases of HCC, so it is not surprising that many of the features show a significant correlation with the target variable. I have included visualizations of the top two most correlated features (in terms of linear correlation, as calculated by F-statistic) below.

In Figure 1, we can see the relationship between performance status and survival rate. Performance status refers to the ECOG Performance Status Scale, a scale used to represent a cancer patient's ability to independently perform daily activities. This feature ranges from 0 to 4, indicating a status of 'active', 'restricted', 'ambulatory', 'selfcare', and 'disabled', respectively. As we might expect, patients with higher scores (lower degree of mobility) had lower survival rates. In Figure 2, we can see the distribution of alkaline phosphatase (ALP) levels, broken out by class. ALP is a liver enzyme with a normal range of 44-147 U/L. This figure clearly shows that more of the class 1 patients had ALP levels

within the normal range, and class 0 patients tended to have higher than normal ALP levels. This finding aligns with medical interpretations of ALP.

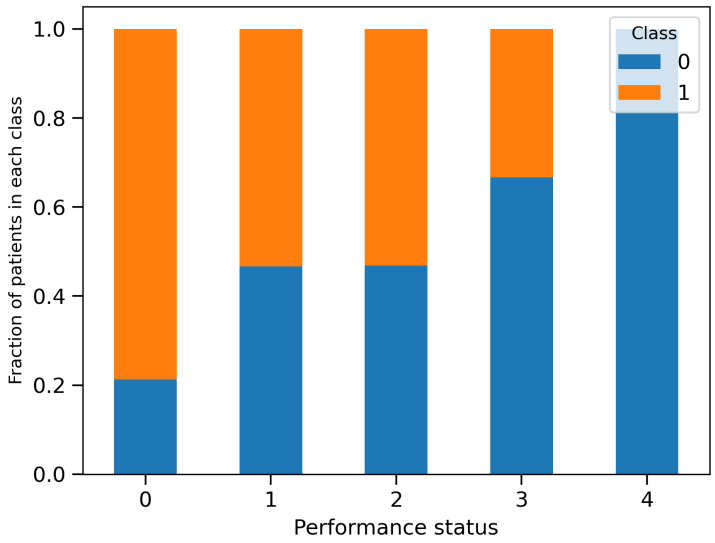

Figure 1: Bar plot illustrating the relationship between ECOG performance status and class. Patients with higher performance statuses (i.e. patients who are less able to perform daily activities) were much more likely to be in class 0.

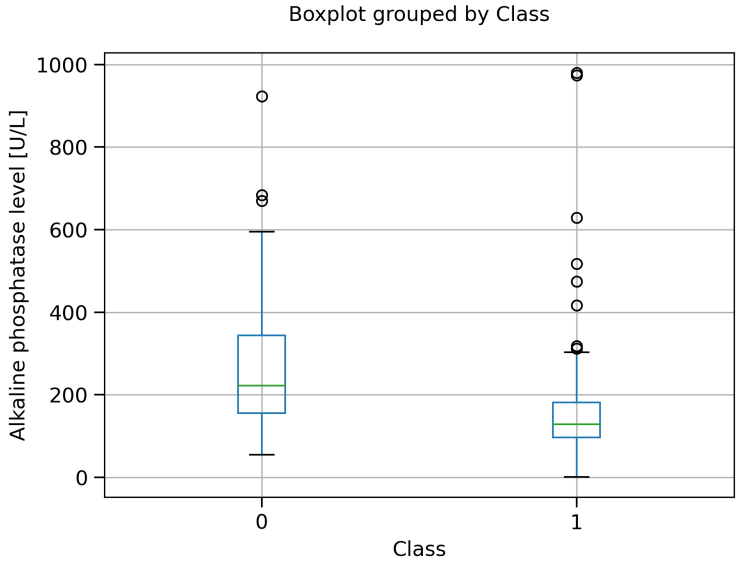

Figure 2: Boxplot showing the distribution of ALP levels in the data set, grouped by class. Class 1 patients tended to have ALP levels closer to the normal range, while patients in class 0 had higher levels of ALP, on average.

Methods

Since this data set has many missing values, I could not use “simpler” algorithms like logistic regression or random forest out of the box. I developed two pipelines: one for XGBoost models (which can accept features with missing values), and one for reduced-feature modeling, which I used to fit models for logistic regression (using L1, L2, and elastic net penalties), random forest, K nearest neighbors, and SVC algorithms.

The XGBoost pipeline is straightforward: I used a simple train-test-split once to create a test set of 20% of the data points, and again to create validation and training sets of 20% and 60%. The data is IID, and the target variable’s class imbalance is relatively minor so I did not need to use stratification or grouping when splitting. I then preprocessed the data. The ordinal features were already formatted to a numerical scale, and did not require any additional preprocessing. The categorical features were binary values (0 or 1), which would not normally require any additional pre-processing. However, the majority of these features contained missing values, which I treated as multi-value features: a patient can have a value of ‘0’, ‘1’ or ‘Unknown’. I therefore processed categorical features using a one-hot encoder. I scaled continuous features using a standard scaler to help the model converge faster. I then fit the models using the parameter values in Table 1, and returned the best model. I repeated this for 10 random states to capture the uncertainty due to splitting and the random seed.

For my reduced feature pipeline, I used train-test-split to create a test set of 20% of the data. Since this method requires fitting a model for each pattern of missing values in the test set, I reduced the number of features missing values by imputing missing values in the discrete features using an unused constant value. This imputation does not affect the distribution of the data; I am using a placeholder to represent a category of ‘Unknown’. For each pattern of remaining missing values, I identified the corresponding test set and training/validation set, and used GridSearchCV to tune hyperparameters. In the pipeline, I used a k-fold split with 4 folds to create the training and validation sets. I processed the categorical features using a one-hot encoder. I then scaled *all* features using a standard scaler. There were two reasons for this: 1) For logistic regression models, scaling all features makes the model more interpretable; 2) Each submodel uses a different combination of continuous features, so the pipeline is much simpler when continuous features are not listed out in the preprocessor. I used the parameter values in Table 1 to fit the models, and returned the best submodels for each pattern. I repeated this process for 10 random states to capture uncertainty due to splitting.

Since the data set is only slightly imbalanced, I used accuracy as the evaluation metric when training the models. Additionally, accuracy was used in both studies mentioned above, so this allows me to compare my models’ performance directly with their results.

Table 1: Hyperparameter values

Algorithm	Parameter Values
Logistic Regression (L1)	C: 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2 Class weight: 'balanced', None
Logistic Regression (L2)	C: 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2 Class weight: 'balanced', None
Logistic Regression (Elastic Net)	C: 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2 Class weight: 'balanced', None L1 ratio: 0.25, 0.5, 0.75
Random Forest	Max depth: 1, 3, 10, 30, 100 Max features: 0.25, 0.5, 0.75, 1 Class weight: 'balanced', None
SVC	C: 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3 Gamma: 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1 Class weight: 'balanced', None
K Nearest Neighbors	n_neighbors: 3, 5, 10 weights: 'uniform', 'distance'
XGBoost	Learning rate: 0.01, 0.03, .05, 0.1, 0.2, 0.3 n_estimators: 10000 (early stopping) reg_alpha: 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2 reg_lambda: 1e-3 , 1e-2, 1e-1, 1e0, 1e1, 1e2 max_depth: 1, 3, 6, 10, 30 colsample_bytree: 0.6, 0.7, 0.8, 0.9 subsample: 0.5, 0.66, 0.75

Results

As shown in Figure 3, all models exceeded (on average) the mean baseline accuracy score of 0.621. However, only the XGBoost, SVC, and L2 logistic regression models had mean accuracy scores more than one standard deviation above the baseline score, and only the XGBoost models had a mean accuracy more than two standard deviations above the baseline. This means that the other four models did not perform significantly better than the baseline. The XGBoost models were the most predictive, with the highest mean accuracy score (0.703) and one of the lowest standard deviations. Additionally, the XGBoost models were the closest to the mean baseline F1 score (0.762), with a mean F1 score of 0.761. See Table 2 for complete results. Compared to the results in the previous papers using this data, the XGBoost models performed about the same as the models developed by Santos et al., but much worse than those developed by Książek et al.

The relative importance of features in the XGBoost models is extremely consistent, both across different calculations of importance, and also with the feature correlations calculated during EDA. As seen in Figures 4-6, global feature importance calculations through SHAP values, permutation, and the XGBoost gain formula all return the same two features as the most important: alpha-fetoprotein and hemoglobin. All three calculations include the same features in the top six most important: alpha-fetoprotein, hemoglobin, iron, ferritin, performance status, alkaline phosphatase, and aspartate transaminase (order varies slightly). This aligns with F-statistic and MI calculations estimating the linear and non-linear correlations between features and the target variable; all six of those features can be found in the top four most correlated features for one of these correlation types. Gender, HIV status, splenomegaly and portal vein thrombosis were all found to be unimportant features.

Figure 7 shows an example of the SHAP local feature importance for a data point in class 0 using one of the XGBoost models. This visualization would be an important tool for clinicians; a doctor can clearly see how test results are contributing to the model's prognosis and can then discuss risk factors with the patient.

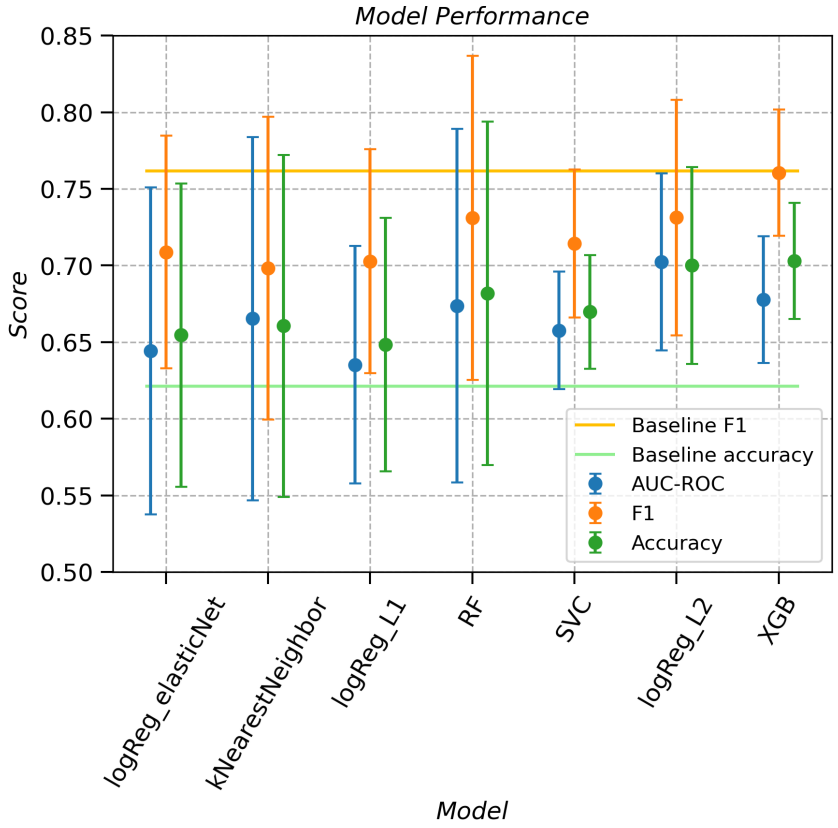

Figure 3: Plot of performance metrics (mean and standard deviation) for each model type. The XGBoost models were the most predictive, with the highest accuracy and F1 scores and the second highest AUC-ROC score.

Table 2: Results

Model	Accuracy Mean	Accuracy Std Dev	F1 Mean	F1 Std Dev	AUC-ROC Mean	AUC-ROC Std Dev
Logistic Regression (L1)	0.648	0.083	0.703	0.073	0.635	0.077
Logistic Regression (L2)	0.700	0.064	0.731	0.077	0.702	0.058
Logistic Regression (EN)	0.655	0.099	0.709	0.076	0.644	0.107
Random Forest	0.682	0.112	0.731	0.106	0.674	0.115
SVC	0.670	0.037	0.714	0.048	0.658	0.038
K Nearest Neighbors	0.661	0.112	0.698	0.099	0.665	0.118
XGBoost	0.703	0.038	0.761	0.041	0.678	0.041
Baseline	0.621	0.065	0.762	0.055	N/A	N/A

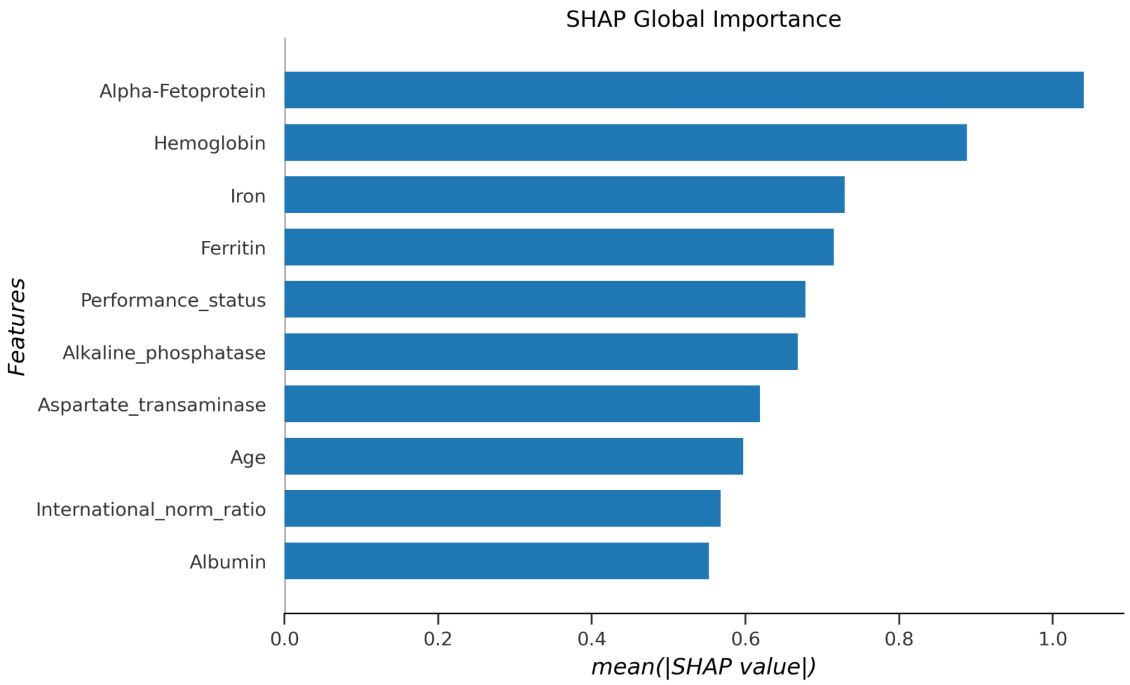

Figure 4: Plot of top ten features by SHAP global importance.

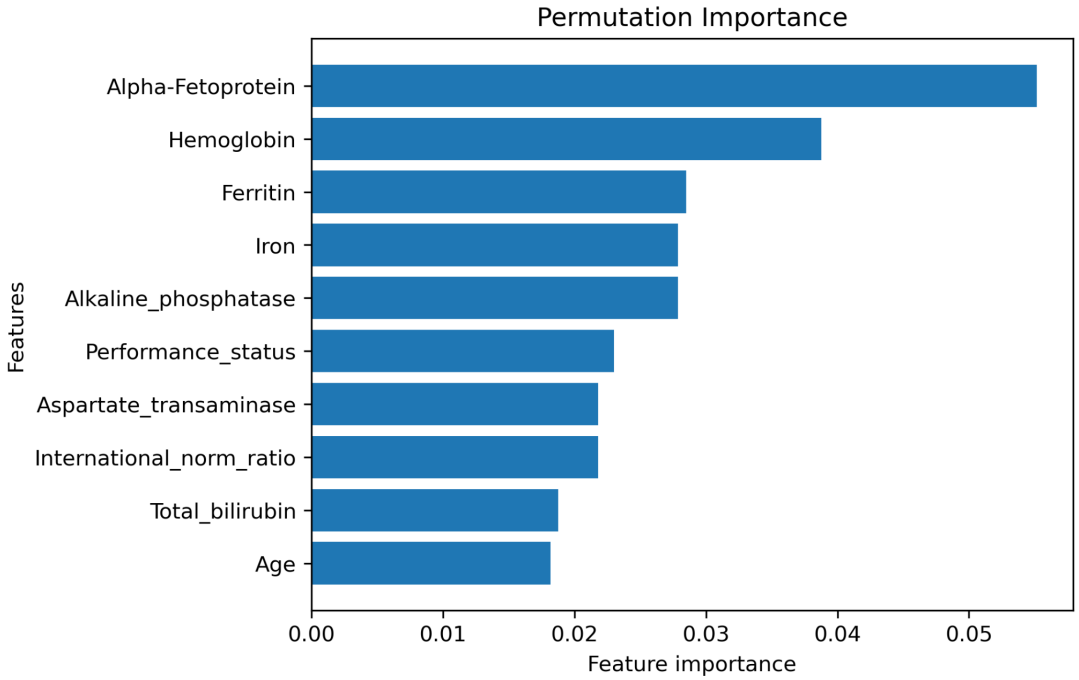

Figure 5: Plot of top ten features by permutation importance.

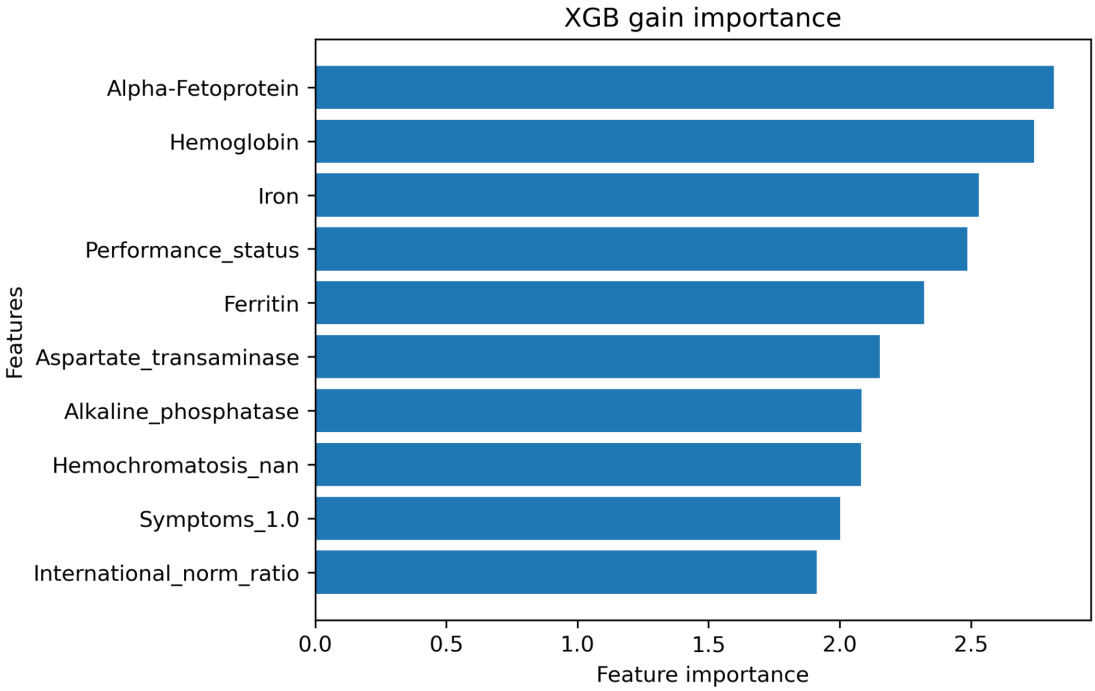

Figure 6: Plot of top ten features by XGBoost gain importance.

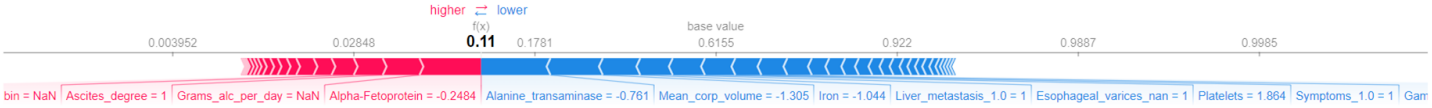

Figure 7: Local SHAP importance for class 0 point. Visualizations like this allow a clinician to interpret the model for a patient. We can see that the patient's low alpha-fetoprotein levels are pushing their survival probability up, while their low iron and alanine transaminase levels and low MCV are pushing their survival probability down.

.

Outlook

As noted in the Results section, most of the models I trained performed poorly, and even the best models did not perform well enough to be used in a clinical setting. The best way to improve the models' accuracy would be to collect more data. A larger data set would reduce variation caused by randomness in splitting, and allow models to identify significant patterns in the data more clearly. This data set was collected from patients in a single hospital; a model like this would need to be trained on a much more diverse data set, with data collected from many hospitals in multiple regions of the world, before it could be considered for wide deployment. I would also confer with an industry expert to ensure that accuracy is the best evaluation metric, or whether an F-beta score giving more weight to either precision or recall would be more appropriate.

Because of the imputed values in the reduced-feature models, the names of one-hot encoded features that contained missing values are non-intuitive, which reduces the interpretability of these models. In addition, there are a number of highly correlated features in the data set, both in the original features and created through preprocessing. Highly correlated features can cause feature importance calculations to underestimate the importance of those features. To make these calculations more reliable and possibly improve model performance, I could try dropping one of each pair of highly correlated features.

References

Santos, Miriam Seoane, Pedro Henriques Abreu, Pedro J Garcia-Laencina, Adelia Simao, Armando Carvalho. 2015. “A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients”, *Journal of Biomedical Informatics* 58: 49-59. Accessed October 6, 2022 from UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/HCC+Survival

Książek, Wojciech, Moloud Abdar, U. Rajendra Acharya, Paweł Pławiak. 2019. “A novel machine learning approach for early detection of hepatocellular carcinoma patients”, *Cognitive Systems Research* 54: 116-127. Accessed October 19, 2022.
https://doi.org/10.1016/j.cogsys.2018.12.001

World Health Organization. “Fact Sheets: Cancer”. February 3, 2022. Accessed October 16, 2022.
https://www.who.int/news-room/fact-sheets/detail/cancer

Mount Sinai Health Library. Accessed October 16, 2022.
https://www.mountsinai.org/health-library/tests/alp-blood-test
https://www.mountsinai.org/health-library/diseases-conditions/ascites