Cluster-Based Oversampling Method for Improving Survival Prediction of Hepatocellular Carcinoma Patients - Revisited

Darwin Agunos

Liver cancer is the sixth most frequently diagnosed cancer. According to the American Cancer Society, it is estimated that there will be 30,000 liver cancer related deaths and 40,000 new cases diagnosed. Hepatocellular Carcinoma (HCC) represents more than 90% of primary liver cancer cancers. Clinicians assess patient treatment based on previously diagnosed cases, which may not always apply to a specific patient given the genetic heterogeneity of the human population. Over the years, research studies have been developing strategies for assisting clinicians in decision making using machine learning techniques to extract knowledge from clinical data. This paper aims to add to the previous methodology of using cluster-based sampling to improve HCC survival prediction models by replicating the original work and incorporating different imputation techniques and classification algorithms. Our results are evaluated in terms of survival prediction. Our approach yielded better prediction scores previously reported, suggesting an improvement over the previous methodology used in Hepatocellular Carcinoma prediction models.

1. Introduction

Solely in 2012, the World Health Organization (WHO) reported about 8.2 million cancer-related deaths and 14.1 million new cancer cases [1]. Also in 2012, the American Cancer Society (ACS) stated that liver cancer is the 3rd leading cause of all cancer related deaths world-wide, 9th in the United States [2]. In the United States 2020, the ACS estimates that there will be 30,000 liver cancer related deaths and 40,000 new cases diagnosed.

Liver cancer trends have been going up for the past forty years in America. Affliction rates have tripled since the 1980's, increasing steadily by 2% annually from mid-2000's to the latter part of the last decade. Fatality rates have also doubled in the past couple decades going from three deaths per 100,000 standard US Population to well over six. However, fatality rates seem to be stabilizing for recent years. The 5-year survival rate for liver cancer is 18% or 18/100 patients.

This paper focuses on a specific type of liver cancer, Hepatocellular Carcinoma. Hepatocellular Carcinoma, or HCC, is responsible for more than 90% of primary liver cancers and it is a major global health problem [3]. It is found in the epithelial cells of the liver.

Data-driven statistical research has become an attractive complement for clinical research. Survival prediction has become one of the most challenging tasks addressed by the medical research community [4-8]. This research consists of analyzing a substantial amount of clinical data, drawings patterns and conclusions from the data and using these inferences to determine the survivability of *X* suffering from *Y* disease over a given *Z* time period. However, modeling and prediction can prove to be a difficult task due to two primary reasons: dataset size and dataset complexity.

Regarding dataset size. Clinical data in the medical field has been localized to small datasets for a variety of reasons. These small datasets limit the scope of data mining techniques because they may not provide adequate information for the algorithms to learn the underlying patterns in the data. [9,10] However, in real-life problems, small datasets are normal.

Data complexity is derived from the characteristics that compose the dataset. For datasets with heterogeneous data, the assumptions of some data mining algorithms may not be verified, thus these algorithms are inapplicable to the task. For datasets with Missing Data (MD), data mining algorithms may produce biased training models and estimates leading to a decrease in evaluation performance. [11]

The aim of this work is to start from the previously published literature regarding the application of computational techniques for HCC and assess their technique to predict patient survival and improve on their techniques if possible. [12] HCC dataset was obtained at a University Hospital in Portugal and contains several demographics, risk factors, laboratory and overall survival features of 165 real patients diagnosed with HCC. The dataset contains 49 predictive variables selected according to the EASL-EORTC (European Association for the Study of the Liver - European Organization for Research and Treatment of Cancer) Clinical

Practice Guidelines, which are the current state-of-the-art on the management of HCC. This dataset also contains a high percentage of missing values (an overall MD rate of 10.22% with only eight patients having complete information. Heterogeneity between patients present due to range of values in the considered variables and presence of class imbalance. Characterization of data could be found in Table 1.

Previous literature regarding computational approaches on HCC are majority based on Neural Networks (NN) and Logistic Regression (LR) models. Most of these works ignore patient heterogeneity and the presence of missing data. The most recent previous work made use of different algorithms to impute missing data and increase NN/LR survival prediction model performance. More specifically, four approaches to increase survival prediction performance where approach three and four introduced a new methodology using cluster-based oversampling.

1. Prediction models directly used after the data imputation phase.
2. Prediction models used after the data imputation phase and oversampled using SMOTE (Synthetic Minority Over-sampling Technique) algorithm [13].
3. Creation of a unique representative dataset using cluster-based oversampling.
4. Creation of several unique representative datasets using cluster-based oversampling where final classification results achieved through majority voting.

Performance indicators for each approach were Accuracy, Area under the ROC Curve (AUC) (Fig.1) and F1 score.

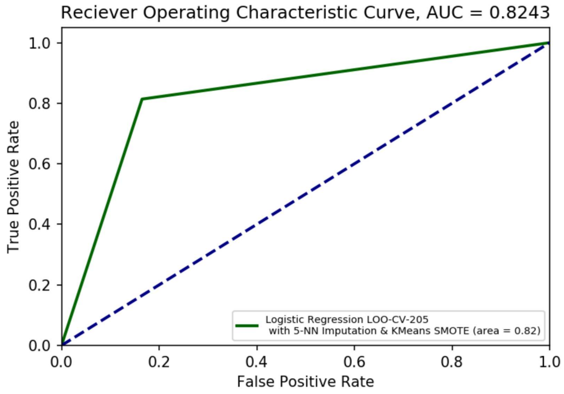


Fig. 1. Area Under the Curve Graph for a LR model using LOO-CV-205 w/ 5NN Imputation & KMeans SMOTE

This paper focuses on re-visiting the previous literature and improving on their techniques. Mainly, we will talk about exploring different machine learning algorithms and imputations techniques introduced in a recent paper [14] to see if we can use this new methodology to improve past results.

Our approaches are tested for both previous data mining algorithms (NN and LR) and new data mining algorithms (Random Forest (RF) , Support Vector Machine (SVM) and XGBoost) using a Leave-One-Out Cross Validation (LOO-CV) approach, which is appropriate for small sample data-sets. For each approach we consider different imputation methods such as Listwise Deletion, Nearest Neighbors using HEOM and Mean/Mode imputation.

Regarding Accuracy, Under the ROC Curve (AUC) and F1-score as performance indicators, our obtained results indicate that cluster-based oversampling using a 5 Nearest Neighbors approach yield the best results over all imputation methods.

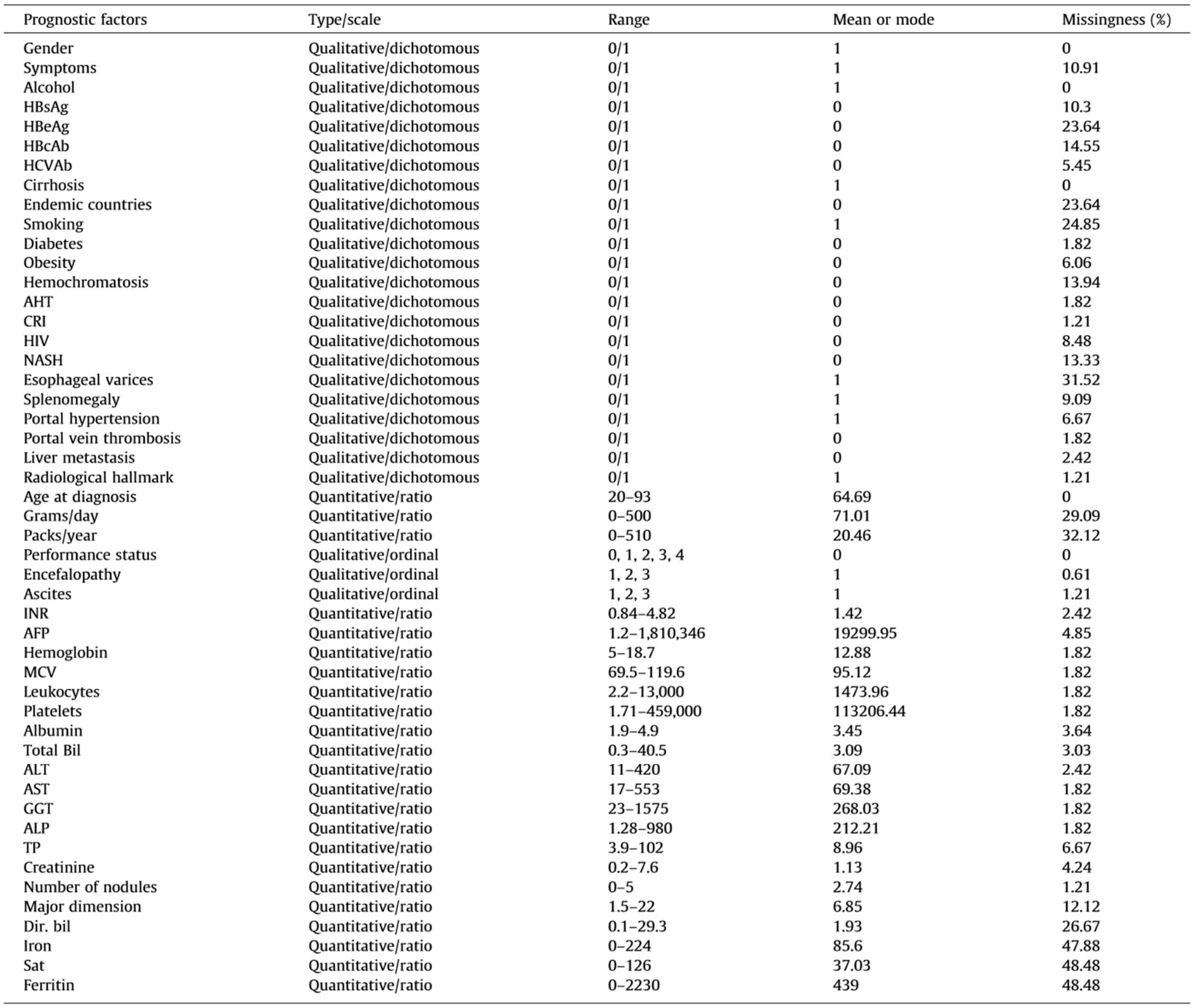
Let's talk about Accuracy, Under the ROC Curve and F1 Score and how we can use them as performance indicators. We assume reader familiarity with a confusion matrix (Fig 2.) and the basic concepts.

Accuracy measures how many observations, both positive and negatives, were correctly classified. It is the sum of true positives and true negatives divided over true positives, false positives, true negatives and false negatives. Accuracy should not be used as the only metric on imbalanced problems. It is easy to get a high accuracy score by simply classifying all observations as the majority class. The equation is as follows:

$$Acc = \frac{tp + tn}{tp + fp + tn + fn} \tag{1}$$

F1 score is a combination of precision and recall into one metric by calculating the harmonic mean between the two. The equation is as follows:

$$F1 = \frac{2\ Precision * Recall}{Precision + Recall} \tag{2}$$



Prognostic factors	Type/scale	Range	Mean or mode	Missingness (%)
Gender	Qualitative/dichotomous	0/1	1	0
Symptoms	Qualitative/dichotomous	0/1	1	10.91
Alcohol	Qualitative/dichotomous	0/1	1	0
HBsAg	Qualitative/dichotomous	0/1	0	10.3
HBeAg	Qualitative/dichotomous	0/1	0	23.64
HBcAb	Qualitative/dichotomous	0/1	0	14.55
HCVAb	Qualitative/dichotomous	0/1	0	5.45
Cirrhosis	Qualitative/dichotomous	0/1	1	0
Endemic countries	Qualitative/dichotomous	0/1	0	23.64
Smoking	Qualitative/dichotomous	0/1	1	24.85
Diabetes	Qualitative/dichotomous	0/1	0	1.82
Obesity	Qualitative/dichotomous	0/1	0	6.06
Hemochromatosis	Qualitative/dichotomous	0/1	0	13.94
AHT	Qualitative/dichotomous	0/1	0	1.82
CRI	Qualitative/dichotomous	0/1	0	1.21
HIV	Qualitative/dichotomous	0/1	0	8.48
NASH	Qualitative/dichotomous	0/1	0	13.33
Esophageal varices	Qualitative/dichotomous	0/1	1	31.52
Splenomegaly	Qualitative/dichotomous	0/1	1	9.09
Portal hypertension	Qualitative/dichotomous	0/1	1	6.67
Portal vein thrombosis	Qualitative/dichotomous	0/1	0	1.82
Liver metastasis	Qualitative/dichotomous	0/1	0	2.42
Radiological hallmark	Qualitative/dichotomous	0/1	1	1.21
Age at diagnosis	Quantitative/ratio	20–93	64.69	0
Grams/day	Quantitative/ratio	0–500	71.01	29.09
Packs/year	Quantitative/ratio	0–510	20.46	32.12
Performance status	Qualitative/ordinal	0, 1, 2, 3, 4	0	0
Encefalopathy	Qualitative/ordinal	1, 2, 3	1	0.61
Ascites	Qualitative/ordinal	1, 2, 3	1	1.21
INR	Quantitative/ratio	0.84–4.82	1.42	2.42
AFP	Quantitative/ratio	1.2–1,810,346	19299.95	4.85
Hemoglobin	Quantitative/ratio	5–18.7	12.88	1.82
MCV	Quantitative/ratio	69.5–119.6	95.12	1.82
Leukocytes	Quantitative/ratio	2.2–13,000	1473.96	1.82
Platelets	Quantitative/ratio	1.71–459,000	113206.44	1.82
Albumin	Quantitative/ratio	1.9–4.9	3.45	3.64
Total Bil	Quantitative/ratio	0.3–40.5	3.09	3.03
ALT	Quantitative/ratio	11–420	67.09	2.42
AST	Quantitative/ratio	17–553	69.38	1.82
GGT	Quantitative/ratio	23–1575	268.03	1.82
ALP	Quantitative/ratio	1.28–980	212.21	1.82
TP	Quantitative/ratio	3.9–102	8.96	6.67
Creatinine	Quantitative/ratio	0.2–7.6	1.13	4.24
Number of nodules	Quantitative/ratio	0–5	2.74	1.21
Major dimension	Quantitative/ratio	1.5–22	6.85	12.12
Dir. bil	Quantitative/ratio	0.1–29.3	1.93	26.67
Iron	Quantitative/ratio	0–224	85.6	47.88
Sat	Quantitative/ratio	0–126	37.03	48.48
Ferritin	Quantitative/ratio	0–2230	439	48.48

Table 1: Characterization of CHUC's hepatocellular carcinoma data. Contains $N = 165$ records of $n = 49$ clinical variables. These variables were considered important to the clinician's decision process. [12]

AUC means area under the curve so to speak about ROC AUC score we need to first define ROC curve first. The ROC curve is a chart that visualizes the tradeoff between the true positive rate (TPR) and false positive rate (FPR). It is the bend present in the upper left corner of Fig. 1. Classifiers that have curves that are more top-left-side are better.

The Area Under the ROC Curve or ROC AUC score is a single number measuring how much space under the curve is encapsulated. It tells us how confident at ranking predictions our model is.

The remainder of this paper is organized as follows: Section 2 presents a brief description about HCC disease and illustrates the previous papers referenced to build upon our work. Section 3 outlines the methodological steps used in this project. Section 4 concerns the experiments. In Section 5 we discuss our results. In Section 6 we present our conclusions, limitations and references to future work.

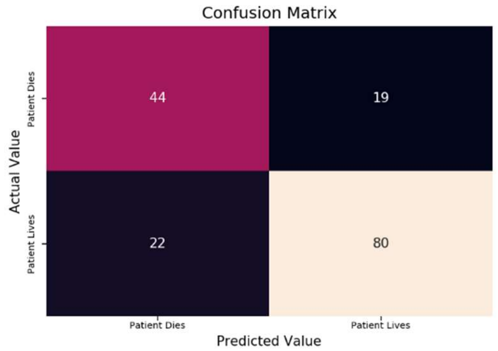


Fig. 2. Confusion Matrix for a XGBClassifier model using LOO-CV-165 with 5-NN Imputation.

2. Computational approached for HCC

In order to predict 1-year survival of HCC patients, it is important to understand some aspects of the pathology and to review previous related works on the application of computational methods to healthcare.

2.1. Notions of HCC disease

A Carcinoma refers to a type of cancer arising from an epithelial cell undergoing a malignant transformation. When the source of cancer is an epithelial cell in the liver, the cancer is referred to as hepatocellular carcinoma (HCC). HCC can have different growth patterns. There is no official cause of HCC, however some proposed causes include:

- Genetic Mutation
- Chronic Infection w/ Hepatitis B/Hepatitis C

Approximately 90% of HCCs are associated with a known underlying risk factor. The most frequent factors include chronic viral hepatitis (types B and/or C) and cirrhosis. Cirrhosis is present in over 80% of HCC cases which clearly identifies it as the main precursor lesion. Risk factors HCC include:

- Being a male (men are three times as likely to be afflicted with liver cancer than women)
- Cirrhosis
- Excessive drinking
- Hepatitis B/Hepatitis C
- High iron levels
- Consumption of Aflatoxin (toxic compound found the molds of certain foods)
- Diabetes
- Accumulation of fatty cells around the liver.

Diagnostic tests include:

- Liver biopsy
- Liver functioning tests
- CAT Scan/MRI

2.2. Previous Works

This paper is a combination of methods presented in two different papers [12,14] to improve predictions on 1-year survival of HCC patients.

1.

In this work published in the *Journal of Biomedical Informatics* a new methodology of predicting 1-year survival was presented. [12] The study presented three main challenges: heterogeneity concerning the type of variables used (49 features with a mixture of dichotomous, ratio scaled and ordinal features), percentage of MD present (overall, MD constituted 10.22% of the total data with only 8/165 patients having complete information), and data imbalance. These challenges make it difficult to create a valid methodology to predict patient 1-year survival. The proposed methodology introduced a cluster-based oversampling approach after data imputation. This methodology, paired with a NN classifier, presented better results than previous literature proving that cluster-based oversampling is a more appropriate approach to design survival prediction models in an HCC context. We base much of our work on this paper.

2.

In this work published in the *Harvard Data Science Review* a new methodology to deal with MD in medical datasets to estimate the probability of drug success through clinical trials was presented. [14] The study used different imputation techniques (Multiple Imputation, Nearest Neighbor, Mean/Mode, Median/Mode and Listwise deletion) in conjunction with six different prediction models (Penalized LR, RF, NN, SVM, Gradient Boosted Trees, and C5.0.) Overall, Nearest Neighbor and Multiple Imputation methods give the best model performance in predicting drug prediction. More specifically, they found the 5NN-RF model gave one of the highest AUC's across all test sets. From this paper, we take their MD imputation methodologies and apply them to HCC survival prediction.

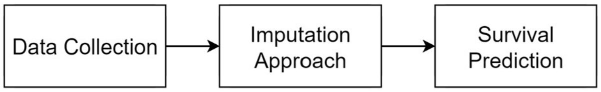


Fig. 3. Proposed methodology

3. Methodology

This section describes the three different stages that compose our improved methodology (Fig. 3.): Data collection, Imputation approaches, Survival

Predictions. The main aspects of each stage are analyzed below.

3.1. Data Collection

The first stage of this methodology has already been performed by the Service of Internal Medicine A of the Coimbra's Hospital and University Centre (CHUC). It concerns the analysis of demographic, risk factor, laboratory and overall survival features from N = 165 patients diagnosed with HCC. The dataset is comprised of n = 49 features selected according to the EASL-EORTC (European Association for the Study of the Liver – European Organization for Research and Treatment of Cancer) Clinical Practice Guidelines [3], the current state-of-the-art on the management of HCC, in collaboration with a team of clinicians from CHUC's Service of Internal Medicine A. The clinical features are considered the most significant to the clinicians' decision process when choosing the most suitable therapeutic strategies and prediction outcomes for patients. The detailed description of the HCC dataset (Table 1) shows each feature's type/scale, range, statistics and missing rate percentage. This is a heterogeneous dataset with twenty-three ratio scaled quantitative variables and twenty- six qualitative variables. Overall, missing data constituted 10.22% of the whole dataset and only 8/165 patients have complete information for all field (4.85%)

The survival target variable has been encoded as a binary variable with values 0 and 1, which respectively means that a patient did not survive or survived. This work focuses on the 1-year survivability prediction on HCC patients with a class distribution (shown in Fig. 4.) of 63 dead (labeled 0) and 102 alive (labeled as 1).

3.2. Imputation Approach

This section is where we begin to build on previous work. In our methodology, this stage entails the process of ensuring that there are no inconsistencies present in the data collected in the previous step, i.e. missing values. In particular, the end of this stage is meant to provide a complete and clean dataset aimed at both minimizing the loss of clinical records and distortion of results presented in the later survival prediction stage. To reiterate, we will be focusing on the imputation and sampling methods we take before the prediction stage.

According to the scientific literature, the two most conventional approaches for managing missing data are to delete or impute instances with missing data. The process of deleting instances wit missing data I called listwise deletion. Listwise deletion is generally not recommended as it leads to information loss due to discarding all observations with missing data. This concern is prevalent with our data since 157 of our 165 patients has missing data. Nevertheless, we check model performance using listwise deletion and compare to other imputation approaches.

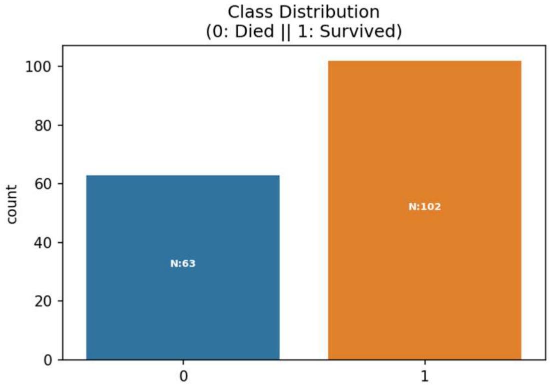


Fig. 4. Class distribution plot with N = 63 for patient died (labeled 0) N = 102 for patient survived (labeled 1)

Since listwise deletion would delete most of our data, an imputation-based approach is considered. Imputation is the process of replacing a missing datum with a substitute value, which is estimated using available information in the dataset. This is an advantage over listwise deletion as imputing missing values provides additional information that will benefit computational methods in predicting patient survival.

In this work we focus on two imputation methods, kNN imputation and Mean/Mode imputation.

In unconditional mean imputation, we fill in the missing values of a variable with the mean/mode of the observed cases of that variable. This method highly discouraged as it can distort the data distribution by reducing variability and undermining the natural relationships between variables. Nevertheless, we implement mean/mode imputation for both categorical and continuous variables and compare to our other approaches.

kNN imputation has shown its usefulness in many other clinical studies with missing values [15-17]. In kNN imputation, given an instance with missing values, we select the k most similar cases without missing values in the selected feature to be imputed. As the name suggests, the replacements for the missing values are chosen from these k nearest neighbors. This allows kNN to maintain the original input data distribution with a proper k selection (we chose default k = 5). For each incomplete case **x**, its closest neighbor **v**, is chosen from an available training sample that has the target variable information. Because of this, the distance between sample **x** and its closest neighbor **v** needs to be computed. The closer the distance, the more similar the instances are. In this work, we use the Heterogeneous Euclidean-Overlap Metric (HEOM) distance [18], which effectively handles both continuous and discrete variables in a missing data framework. Considering two input vectors $\mathbf{x_a}$ and $\mathbf{x_b}$, the HEOM distance can be calculated by

$$\mathrm{HEOM}(x_a, x_b) = \sqrt{\sum_{j=1}^{n} d_j\left(x_{aj}, x_{bj}\right)^2} \qquad (3)$$

where $d_j\left(x_{aj}, x_{bj}\right)$ is the distance between the two cases on its j-th attribute where

$$d_j(x_a j, y_b j) = \begin{cases} 1 & \text{if } x_j \text{ is missing in } x_a \text{ or } x_b \\ d_o(x_a j, y_b j) & \text{if } x_j \text{ is a discrete variable} \\ d_n(x_a j, y_b j) & \text{if } x_j \text{ is a continuous variable} \end{cases} \qquad (4)$$

In the equation above, distance varies from 0 to 1 (the maximal distance value). If either one of the input values is missing in the jth variable, its distance is 1. If both input values are available, HEOM uses the overlap metric, d_o, for categorical attributes and the normalized Euclidean distance, d_n for continuous attributes. These can be found below.

Discrete Data

$$d_o(x_a j, y_b j) = \begin{cases} 1 & \text{if } x_{aj} = x_{bj} \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

Continuous Data

$$d_n(x_a j, y_b j) = \frac{|x_{aj} - x_{bj}|}{\max(x_j)\text{-}\min(x_j)} \qquad (6)$$

Once the closest neighbor **v** is found, each unknown value in **x** is replaced by the corresponding available feature of **v**.

Finally, at the end of the data imputation stage, all features are standardized using the well-known Z-Score transformation.

3.2.1 Sampling Methods

After imputation we can begin to sample our data. Looking at Fig. 4 we can see that we have class imbalance in our dataset. The Class distribution is $N = 63$ for patients who died (labeled 0) $N = 102$ for patients who survived (labeled 1). If we used our original imbalanced dataframe then we would more than likely run into two issues, overfitting and wrong correlation.

Overfitting is the production of an analysis that corresponds too closely or exactly to a set of data and may therefore fail to fit additional data or predict future observations. Overfitting in an imbalanced dataset can lead to a machine learning algorithm "learning" to be biased towards one criterion over another, thus preventing the algorithm to learn the true nature of the data.

If our data is imbalanced this would also lead to incorrect correlation values. We stated earlier that these features were chosen by clinicians based on their importance to liver cancer issues. It would be useful to see how these features could influence our results. By having an imbalanced dataframe we are unable to see the true correlation values between the class and our features.

To correct this, we should create sub-samples of our dataset after the data imputation stage to achieve a dataframe that has a 50/50 ratio between patients who survived and patients who die. Random oversampling and under-sampling are the two most common sampling techniques. Oversampling augments the dataset by replicating random instances from the minority class. Undersampling removes majority class examples from the original set. Both techniques can achieve class balance but can also have their own faults. Oversampling does not add any new information and can result in a machine learning task overfitting the data. Undersampling can lead to loss of important information hindering a machine learning task.

To avoid these drawbacks, we take advantage of the SMOTE algorithm [19], which is the most popular and applied oversampling procedure. The SMOTE algorithm creates synthetic data based on the minority class and the kNN algorithm. This process is shown in Fig. 5.

3.2.2 K-means

Once the data is clean and balanced, we try to find naturally occurring clusters (or groups) within our HCC database. Each group will be composed of several patient samples with similar features. Clustering is performed using the *K*-means algorithm. *K*-means is a well-known unsupervised learning algorithm for data partitioning. *K*-means clustering aims to partition *N* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. The optimal number of clusters found in the previous work was $k = 10$ clusters. This process is shown in Fig. 6.

3.4 Survival Prediction

In the previous work, they combined two well-known classification methods: Neural Networks (NN) and Logistic Regression (LR) in conjunction with a cluster-based oversampling method for patient survival prediction for HCC. In total, each classifier compared performances between four approaches.

In our work, we explore different imputation and sampling methods to compare performances between methodologies and machine learning models. In total, we obtain results between nine different approaches for five different machine learning models (LR, RF, XGBoost, SVM, NN)**.** Performance between approaches are characterized through Accuracy, F1-Score and AUC.

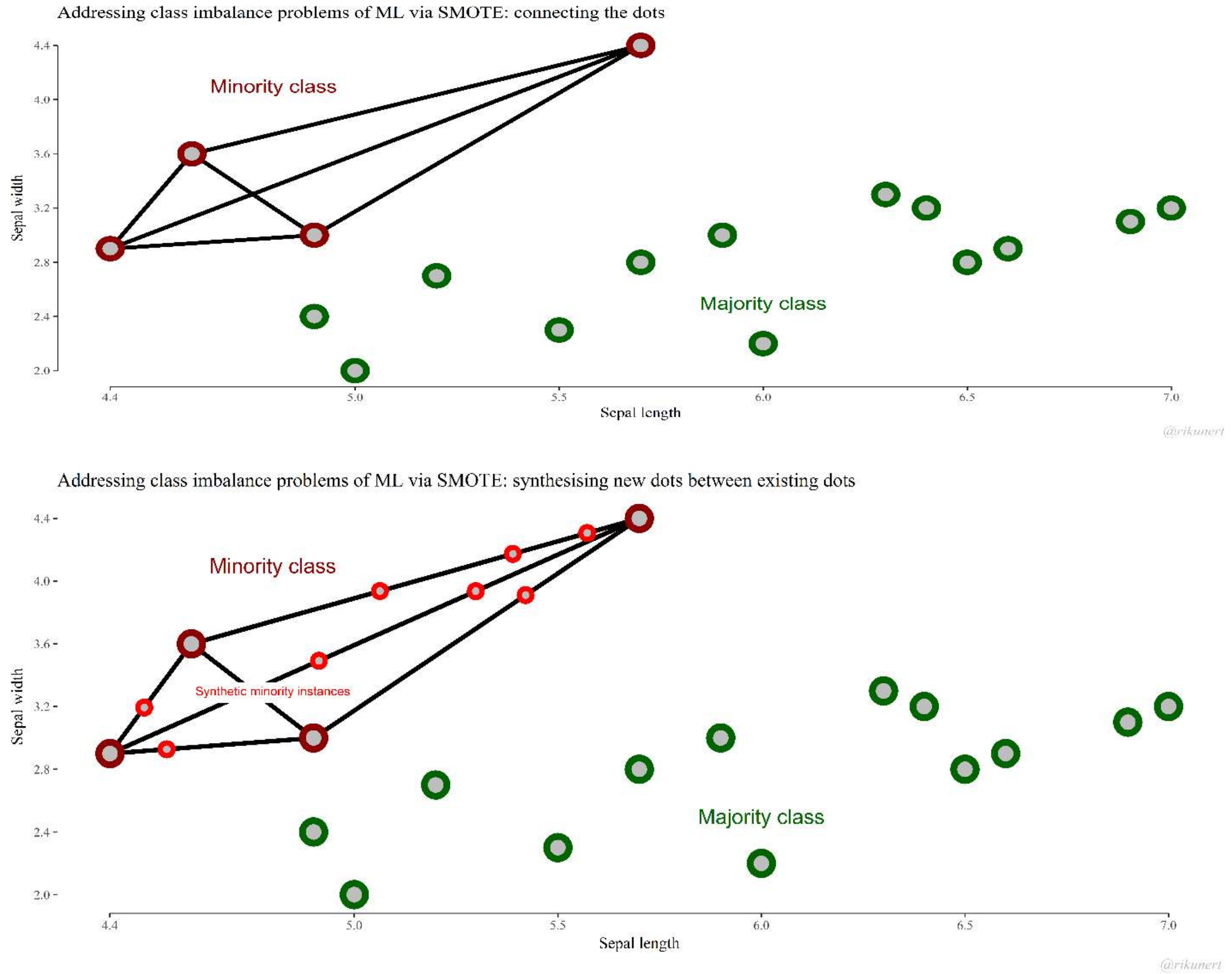
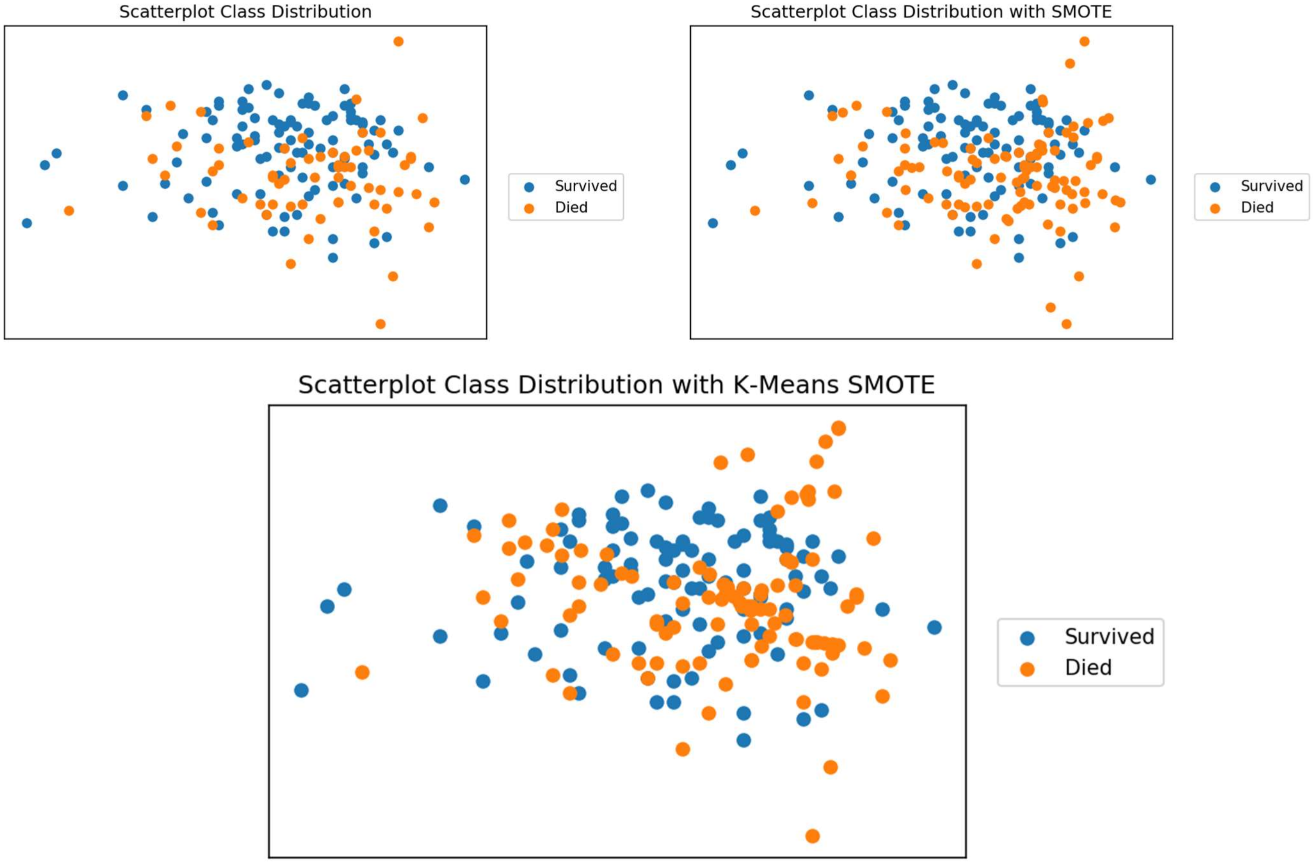


Fig. 5. Synthesis of minority class instances via SMOTE. Photo credits Rikunert.com


Scatterplot Class Distribution

Scatterplot Class Distribution with SMOTE

Scatterplot Class Distribution with K-Means SMOTE

Fig. 6. Scatterplots detailing creation of synthetic samples using SMOTE & K-Means SMOTE.

4. Experiments

The proposed methodology has been experimentally evaluated using the HCC dataset. These experiments were performed in order to replicate and build upon pre-existing literature using different machine learning algorithms and imputation techniques introduced in other works. In this work, we carried out a series of simulations considering nine different approaches. These approaches are detailed below

- Golden Standard
- Golden Standard w/ LOO-CV-8
- 5NN w/ LOO-CV-165
- 5NN w/ LOO-CV-204 & SMOTE
- 5NN w/ LOO-CV-141 & Undersampling
- 5NN w/ LOO-CV-204 & SMOTENC
- 5NN w/ LOO-CV-205 & K-Means SMOTE
- Mean/Mode w/ LOO-CV-165
- Mean/Mode w/ LOO-CV-204 & SMOTE
- Mean/MODE w/ LOO-CV-205 & K-Means SMOTE

Golden Standard refers to listwise deletion of data. 5NN refers to kNN imputation using $k = 5$ neighbors. Mean/Mode refers to the imputation method for categorical and continuous variables in our dataset. For each approach and each classification algorithm, a survival prediction model is constructed.

Experiments performed using a Leave-One-Out Cross Validation (LOO-CV), with the exception of the Golden Standard method for performance evaluation. This approach is appropriate since the amount of available data is not large. This allows for each instance in our dataset to be part of a training and test set. More specifically, for the N total instances involved in our study, one is left out for testing and the remaining $N - 1$ instance are used to design the survival prediction models.

All simulations have been carried out on the same machine in Python 3.7.4 using various popular packages associated with machine learning and scientific analysis (numpy, pandas, matplotlib, scikit-learn, scipy, seaborn, tensorflow, imbalanced learn, etc…).

Performance evaluation for each classifier were based on three different measures: Accuracy, F1-Score and AUC (Table 3). Accuracy is the traditional and most widely-used performance measure in classification problems, however it ignores the probability estimation of classifications in favor of class labels and can be an inappropriate measure in imbalanced problems or problems where you value classification of a certain class more so than the other. To counteract the flaws of basing model performance on Accuracy, two additional performance measures based on the confusion matrix and ROC (Receiver Operating Curve) are applied: F1-score and AUC (area under the ROC curve). F1-score is a combination of Precision and Recall into one metric by calculating the harmonic mean between the two. It provides a balance between both performance metrics; thus, it is a widely used performance measure for regular measurement and imbalanced class problems. The area under the ROC curve (AUC) is a single number representative of how well a classification model can distinguish between two groups. An AUC score of 1 means perfect model prediction while an AUC score of 0 means no correct predictions. Generally, most graphs will have a random predictor, a dashed line across the curve with a AUC score of 0.5. This random predictor is commonly used as a baseline to evaluate model usefulness. In practice, you ideally want the AUC score to be much greater or lower than 0.5. A model trained on clean data and with an AUC < 0.5 just needs to invert the decision the model is making. In the end, we also found the mean scores for all approaches (found in Table 2).

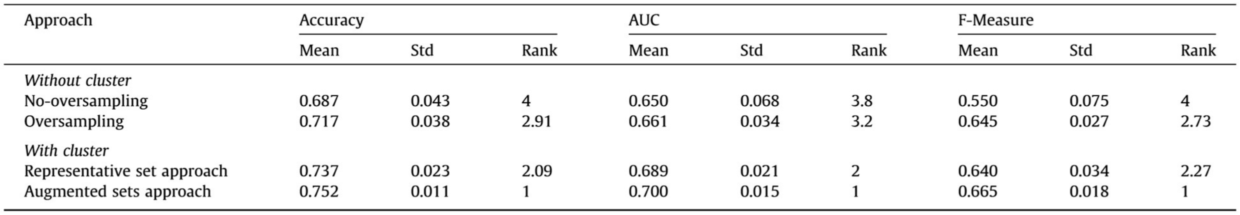
We discuss and compare our results to results found in previous scientific literature [12] (shown in Tables 2-5).

Imputation Method	Machine-learning Model	Metrics for Model Testing Set		
		Acc (%)	F1	AUC
Golden Standard	Mean Machine-learning scores	12.50	0.1250	0.1250
Golden Standard w/ LOO-CV-8		27.50	0.2722	0.2750
5NN w/ LOO-CV-165		70.18	0.6995	0.6775
5NN w/ LOO-CV-204 & SMOTE		78.53	0.7849	0.7853
5NN w/ LOO-CV-204 & Undersampling		72.54	0.7251	0.7223
5NN w/ LOO-CV-204 & SMOTENC		76.67	0.7656	0.7667
5NN w/ LOO-CV-204 & K-Means SMOTE		76.97	0.7695	0.7697
Mean/Mode w/ LOO-CV-165		69.15	0.6914	0.6648
Mean/Mode w/ LOO-CV-204 & SMOTE		77.45	0.7745	0.7745
Mean/Mode w/ LOO-CV-205 & K-Means SMOTE		76.98	0.7695	0.7696

Table 2. Mean performance metrics for all approaches and classifiers

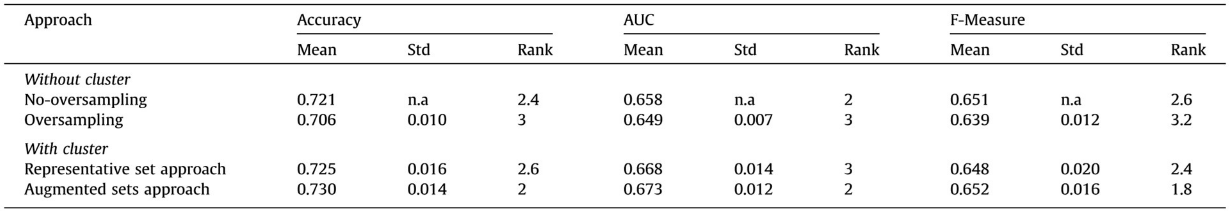
Imputation Method	Machine-learning Model	Metrics for Model Testing Set		
		Acc (%)	F1	AUC
Golden Standard	Logistic Regression	0.00	0.0000	0.0000
Golden Standard w/ LOO-CV-8		12.50	0.1111	0.1250
5NN w/ LOO-CV-165		73.94	0.739	0.7250
5NN w/ LOO-CV-204 & SMOTE		79.90	0.7790	0.7990
5NN w/ LOO-CV-141 & Undersampling		73.36	0.7378	0.7353
5NN w/ LOO-CV-204 & SMOTENC		75.00	0.7499	0.7500
5NN w/ LOO-CV-204 & K-Means SMOTE		79.02	0.7902	0.7902
Mean/Mode w/ LOO-CV-165		73.94	0.7390	0.7225
Mean/Mode w/ LOO-CV-204 & SMOTE		79.41	0.7941	0.7941
Mean/Mode w/ LOO-CV-205 & K-Means SMOTE		78.54	0.7845	0.7854
Golden Standard	Random Forest	0.00	0.00	0.0000
Golden Standard w/ LOO-CV-8		50.00	0.4667	0.5000
5NN w/ LOO-CV-165		66.06	0.6616	0.6436
5NN w/ LOO-CV-204 & SMOTE		77.94	0.7782	0.7794
5NN w/ LOO-CV-141 & Undersampling		73.76	0.7381	0.7369
5NN w/ LOO-CV-204 & SMOTENC		73.04	0.7263	0.7304
5NN w/ LOO-CV-204 & K-Means SMOTE		72.68	0.7258	0.7265
Mean/Mode w/ LOO-CV-165		65.45	0.6551	0.6356
Mean/Mode w/ LOO-CV-204 & SMOTE		74.02	0.7374	0.7402
Mean/Mode w/ LOO-CV-205 & K-Means SMOTE		76.10	0.7605	0.7608
Golden Standard	XGBoost	N/A	N/A	N/A
Golden Standard w/ LOO-CV-8		0.00	0.00	0.00
5NN w/ LOO-CV-165		69.09	0.6856	0.6590
5NN w/ LOO-CV-204 & SMOTE		75.98	0.7595	0.7598
5NN w/ LOO-CV-141 & Undersampling		70.92	0.7069	0.7005
5NN w/ LOO-CV-204 & SMOTENC		81.86	0.8186	0.8186
5NN w/ LOO-CV-204 & K-Means SMOTE		74.15	0.7415	0.7415
Mean/Mode w/ LOO-CV-165		69.70	0.6910	0.6639
Mean/Mode w/ LOO-CV-204 & SMOTE		81.37	0.8136	0.8137
Mean/Mode w/ LOO-CV-205 & K-Means SMOTE		75.12	0.7512	0.7513
Golden Standard	Support Vector Machine	N/A	N/A	N/A
Golden Standard w/ LOO-CV-8		0.00	0.00	0.0000
5NN w/ LOO-CV-165		68.48	0.6815	0.6571
5NN w/ LOO-CV-204 & SMOTE		81.86	0.8181	0.8186
5NN w/ LOO-CV-141 & Undersampling		75.18	0.7524	0.7558
5NN w/ LOO-CV-204 & SMOTENC		79.90	0.7979	0.7990
5NN w/ LOO-CV-204 & K-Means SMOTE		77.07	0.7707	0.7708
Mean/Mode w/ LOO-CV-165		68.18	0.6818	0.6541
Mean/Mode w/ LOO-CV-204 & SMOTE		79.90	0.7986	0.7990
Mean/Mode w/ LOO-CV-205 & K-Means SMOTE		77.56	0.7755	0.7757
Golden Standard	Neural Network	50.00	0.5000	0.5000
Golden Standard w/ LOO-CV-8		75.00	0.7500	0.7500
5NN w/ LOO-CV-165		73.33	0.7293	0.7054
5NN w/ LOO-CV-204 & SMOTE		76.96	0.7695	0.7696
5NN w/ LOO-CV-141 & Undersampling		69.50	0.6905	0.6832
5NN w/ LOO-CV-204 & SMOTENC		73.53	0.7353	0.7353
5NN w/ LOO-CV-204 & K-Means SMOTE		81.95	0.8195	0.8196
Mean/Mode w/ LOO-CV-165		68.48	0.6848	0.6480
Mean/Mode w/ LOO-CV-204 & SMOTE		72.55	0.7254	0.7255
Mean/Mode w/ LOO-CV-205 & K-Means SMOTE		77.56	0.7754	0.7750

Table 3. Performance metrics for each approach and classifier



Approach	Accuracy			AUC			F-Measure		
	Mean	Std	Rank	Mean	Std	Rank	Mean	Std	Rank
Without cluster									
No-oversampling	0.687	0.043	4	0.650	0.068	3.8	0.550	0.075	4
Oversampling	0.717	0.038	2.91	0.661	0.034	3.2	0.645	0.027	2.73
With cluster									
Representative set approach	0.737	0.023	2.09	0.689	0.021	2	0.640	0.034	2.27
Augmented sets approach	0.752	0.011	1	0.700	0.015	1	0.665	0.018	1

Table 4. Neural Networks (NN) performance evaluation from the previous study using Accuracy, AUC and F-Measure. For each measure, three indicators were used: mean and standard deviation (std) for the best configuration of each approach and the Rank of the Friedman rank test. [12]



Approach	Accuracy			AUC			F-Measure		
	Mean	Std	Rank	Mean	Std	Rank	Mean	Std	Rank
Without cluster									
No-oversampling	0.721	n.a	2.4	0.658	n.a	2	0.651	n.a	2.6
Oversampling	0.706	0.010	3	0.649	0.007	3	0.639	0.012	3.2
With cluster									
Representative set approach	0.725	0.016	2.6	0.668	0.014	3	0.648	0.020	2.4
Augmented sets approach	0.730	0.014	2	0.673	0.012	2	0.652	0.016	1.8

Table 5. Logistic Regression (LR) performance evaluation from the previous study using Accuracy, AUC and F-Measure. For each measure, three indicators were used: mean and standard deviation (std) for the best configuration of each approach and the Rank of the Friedman rank test. It should be noted that for Without Cluster and No-Oversampling, std values are not applicable, n.a. [12]

5. Discussion

5.1. Comparing Results Between Papers

The proposed methodology has been experimentally evaluated using the HCC dataset. These experiments were performed in order to replicate and build upon pre-existing literature. More specifically, we compare imputation and sampling methods between our Logistic Regression and Neural Network survival prediction models. From there, we build upon the literature, exploring the performance of different classification algorithms and imputation methods [Table 3].

Overall, it seems that our results have mixed agreement with the original paper. Taking the performance of each imputation method of each classifier and averaging out the scores over the number of classifiers we can obtain the mean score for each approach [Table 2]. In the original paper, cluster-based oversampling (shown as: *with cluster*, Representative set approach) paired with Nearest Neighbor imputation performed better than oversampling and no oversampling without clustering. In our approach, oversampling without clustering performed better when it's score was averaged over all classifiers; however, cluster-based oversampling performed the best in survival predictions when paired with a NN classifier like in the original paper.

In our approach, cluster-based oversampling with Nearest Neighbor imputation is shown as: 5NN w/ LOO-CV-204 & K-Means SMOTE, oversampling without clustering as: 5NN w/ LOO-CV-204 & SMOTE and no oversampling without clustering as: 5NN w/ LOO-CV-165.

Over all approaches, our LR and NN survival prediction models show a substantial increase in scores over all metrics e.g. Our NN paired with a cluster-based oversampling methodology performed approximately 10% better in Accuracy, 15% better in AUC and 16% better in F1-score. These results suggest a massive improvement over the old methodology.

It should be noted that in the original paper they performed thirty runs of over-sampling with each iteration of LOO-CV which we did not perform here. Constant reiterations allowed them to take the standard deviation scores of certain approaches. While our cluster-based oversampling approach performed worse on paper than the oversampling without clustering approach when averaged over each classifier, these results were only taken through one iteration due to computational limitations. Results are subject to change when averaged over many iterations.

In the original paper, utilizing a NN paired with a cluster-based oversampling approach yielded better results than a LR paired with the same methodology. This trend still holds up here.

5.2. New Paper, New Results

In this section we discuss the approaches and classifiers we utilized that are not present in the original paper. More specifically, the different approaches taken towards dealing with missing data (Mean/Mode imputation and listwise deletion) and the performance of the previously mentioned classification algorithms (Logistic Regression, Neural Network) along with three different classification algorithms (Random Forest, XGBoost and Support Vector Machine).

Other than kNN imputation to deal with missing data, we also explored a listwise deletion approach (labeled Golden Standard) and a Mean/Mode imputation-based approach. Taking results from Table 3, the approach that yielded the most optimal results for HCC survival prediction is using a NN paired with the 5NN w/ LOO-CV-204 & K-Means SMOTE approach. kNN based approaches to dealing with missing data, on average, yielded better results than listwise deletion and worse results than mean/mode imputation. Unsurprisingly, the worst approach was listwise deletion which removed nearly 96% of our data.

Except for kNN imputation used in conjunction with SMOTE, kNN imputation performed worse than mean/mode imputation. However, mean/mode imputation can lead to distorted results. It is believed that the noise introduced by mean and median imputation work in adversity against a classifier's learning process. These effects may not seem obvious through testing sets, however it is reported in literature that the noise introduced by the distortion can hinder a machine learning algorithm from capturing the underlying relationships in the data.

Focusing on our most optimal imputation method, 5NN w/ LOO-CV-204 & K-Means SMOTE, the prediction model that performed the best was the Neural Network prediction model. The Logistic Regression prediction model performed second best using this methodology followed by the Support Vector Machine and XGBoost survival prediction models and ending with the Random Forest prediction model performing the worst.

In our approach, cluster-based oversampling with Nearest Neighbor imputation is shown as 5NN w/ LOO-CV-204 & K-Means SMOTE. This approach led to worse performance on HCC survival prediction than oversampling without clustering (detailed in our approach as: 5NN w/ LOO-CV-204 & SMOTE) and better performance than no oversampling without clustering (detailed in our approach as: 5NN w/ LOO-CV-165).

For the approach we report mean metric scores:

- Accuracy: 76.97% (1.56% worse than oversampling without clustering, 6.79% better than no oversampling without clustering)
- F1-score: 0.7695 (1.54% worse than oversampling without clustering, 7% better than no oversampling without clustering)
- AUC: 0.7697 (1.56% worse than oversampling without clustering, 9.22% better than no oversampling without clustering)

6. Conclusions, Limitations and Future Work

In this work we set out to replicate and build upon pre-existing literature for HCC patient survival predictions. To achieve this, we used an HCC dataset composed by 165 patients followed in Coimbra's Hospital and University Centre. In the original paper, we to deal with three main challenges with the dataset: the heterogeneity of variables (49 features chosen by clinicians for its importance in liver cancer encompassing both nominal and continuous traits), the missing data (MD) present in the dataset (overall, MD constituted 10.22% of the total data with only 8/165 patients having complete information across all features) and data imbalance. These issues make it difficult for HCC survival prediction when using machine-learning algorithms.

Our methodology builds upon the results of the previous paper by incorporating different imputation techniques and classifiers. Imputation methods and models were assessed using three performance metrics: Accuracy, F1-score and AUC. Our replicated results agreed with the original paper when applicable. A cluster-based oversampling approach after the data imputation stage yielded better results for the patient survival prediction stage than no oversampling without clustering and oversampling without clustering reiterating that a cluster-based oversampling approach is a more appropriate approach to designing survival prediction models in an HCC context. In comparison between LR and NN classifiers using the proposed methodology, the NN classifier performs better across all metrics, keeping in line with the current literature. Incorporating different classifiers, the Neural Network classifier still yielded the best results when paired with the proposed methodology supporting the idea that Neural Networks are the best classifier to use for survival prediction models in the context of HCC disease.

Limitations with our work is mainly stunted at computational power. All work was performed in Python on the same machine. Performing sampling methods at every LOO-CV iteration was unfeasible. This same limitation applied when we tried using Multiple Imputation as another imputation approach. These limitations are also why we opted to disregard the augmented set approach (approach 4 in the original paper) in our results.

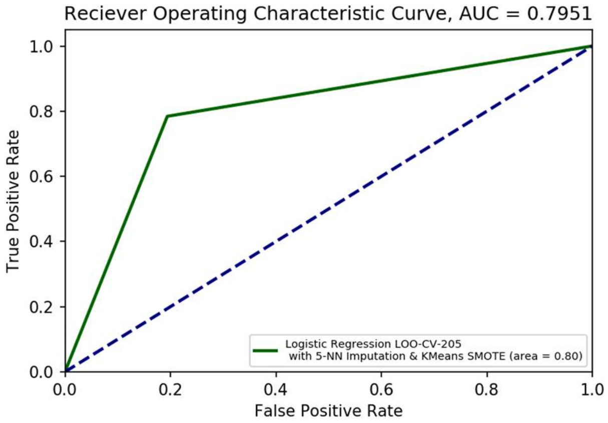
One idea to explore in future work is the use of the Gower Distance as a distance metric between neighbors. It is a distance metric that can handle both categorical and continuous variables.

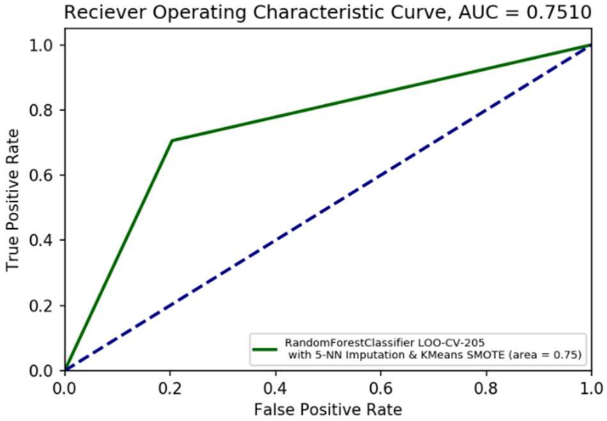
References

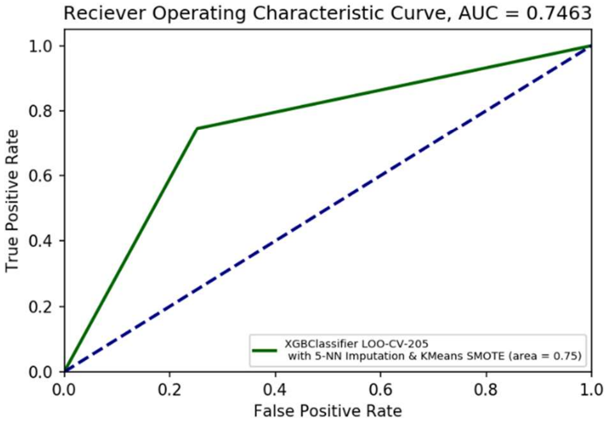
1. W.H. Organization, Globocan 2012: estimated cancer incidence, mortality and prevalence worldwide in 2012. https://http//globocan.iarc.fr/
2. https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/cancer-facts-figures-2012.html
3. Anon., European association for the study of the liver, European organization for research and treatment of cancer, EASL-EORTC clinical practice guidelines: management of hepatocellular carcinoma, J. Hepatol. 56 (4) (2012) 908–943.
4. [H.B. Burke, P.H. Goodman, D.B. Rosen, D.E. Henson, J.N. Weinstein, F.E. Harrell, J.R. Marks, D.P. Winchester, D.G. Bostwick, Artificial neural networks improve the accuracy of cancer survival prediction, Cancer 79 (4) (1997) 857–862.
5. J. Thongkam, G. Xu, Y. Zhang, F. Huang, Toward breast cancer survivability prediction models through improving training space, Expert Syst. Appl. 36 (10) (2009) 12200–12209.
6. N. Esfandiari, M.R. Babavalian, A.-M.E. Moghadam, V.K. Tabar, Knowledge discovery in medicine: current issue and future trend, Expert Syst. Appl. 41 (9) (2014) 4434–4463.
7. P.H. Abreu, H.A. Amaro, D. Castro-Silva, P. Machado, M.H. Abreu, N. Afonso, A. Dourado, Overall survival prediction for women breast cancer using ensemble methods and incomplete clinical data, in: L.M. Roa Romero (Ed.), XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013, IFMBE Proceedings, vol. 41, Springer, 2014, pp. 1366–1369.
8. P.H. Abreu, H.A. Amaro, D. Castro-Silva, P. Machado, M.H. Abreu, N. Afonso, A. Dourado, Personalizing breast cancer patients with heterogeneous data, in: Y.T. Zhang (Ed.), International Conference on Health Informatics, IFMBE Proceedings, vol. 42, Springer, 2014, pp. 39–42.
9. J. Yuan, T. Fine, Neural-network design for small training sets of high dimension, IEEE Trans. Neural Netw. 9 (2) (1998) 266–280.
10. R. Andonie, Extreme data mining: Interference from small datasets, Int. J. Comput. Commun. Control 5 (3) (2010) 280–291.
11. [P.J. García-Laencina, J.L. Sancho-Gómez, A. Figueiras-Vidal, Pattern classification with missing data: a review, Neural Comput. Appl. 19 (2010) 263–282.
12. Miriam Seoane Santos, Pedro Henriques Abreu, Pedro J Garcia-Laencina, Adelia Simao, Armando Carvalho, A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients, Journal of biomedical informatics, 58, 49-59, 2015.
13. N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357.
14. Lo, A. W., Siah, K. W., & Wong, C. H. (2019). Machine Learning with Statistical Imputation for Predicting Drug Approvals. Harvard Data Science Review, 1(1). https://doi.org/10.1162/99608f92.5c5f0525
15. R.J.A. Little, Methods for handling missing values in clinical trials , J.Rheumatol. 26 (8) (1999) 1654–1656.
16. O. Troyanskaya, M. Cantor, O. Alter, G. Sherlock, P. Brown, D. Botstein, R. Tibshirani, T. Hastie, R. Altman, Missing value estimation methods for DNA microarrays, Bioinformatics 17 (2001) 520–525.
17. J.M. Jerez, I. Molina, P.J. Garcia-Laencina, E. Alba, N. Ribelles, M. Martin, L. Franco, Missing data imputation using statistical and machine learning methods in a real breast cancer problem, Artif. Intell. Med. 50 (2) (2010) 105–115.
18. G.E. Batista, M.C. Monard, An analysis of four missing data treatment methods for supervised learning, Appl. Artif. Intell. 17 (2003) 519–533.
19. N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357.

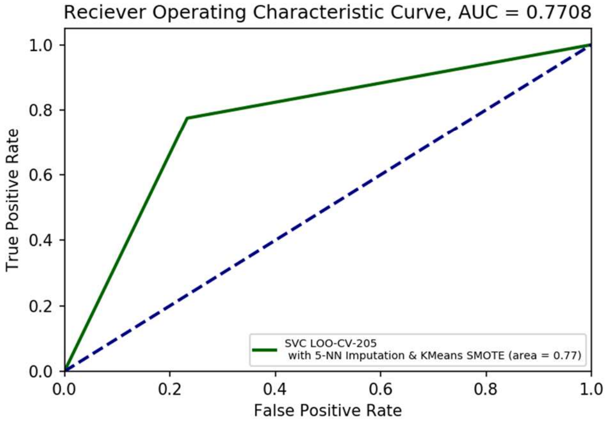
Appendix A:

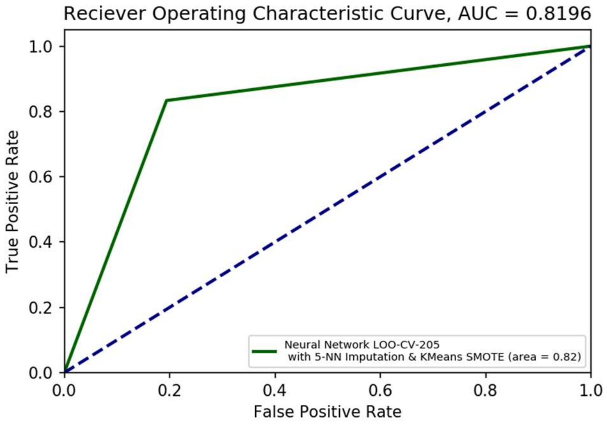
ROC Curve Plots and Confusion Matrixes
for Classification Algorithms utilizing the 5NN w/ LOO-CV-204 & K-Means SMOTE Approach

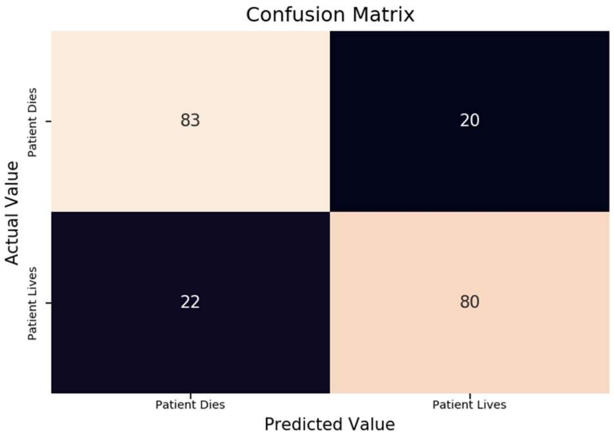




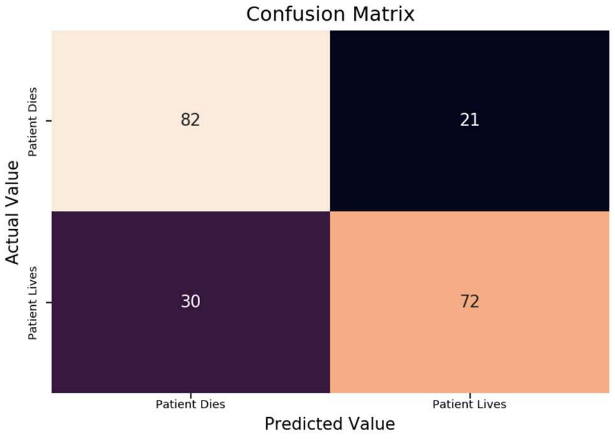




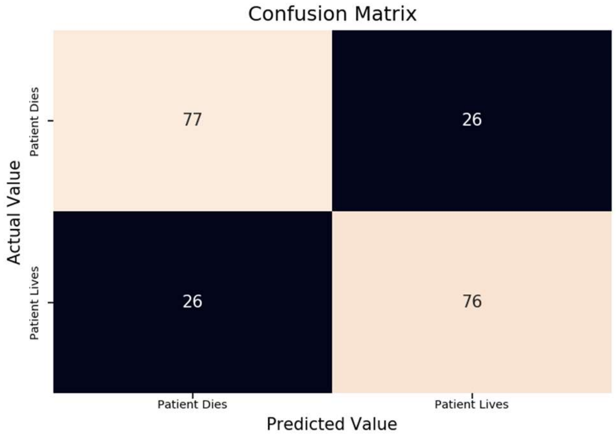




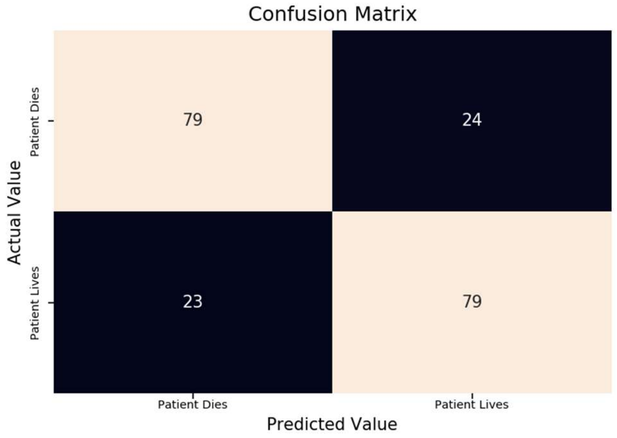
Logistic Regression Confusion Matrix



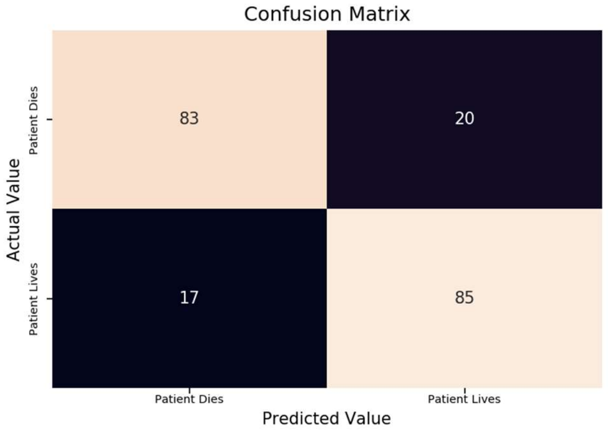
Random Forest Confusion Matrix



XGBoost Confusion Matrix



Support Vector Machine Confusion Matrix



Neural Network Confusion Matrix