Predicting mortality risk in patients suffering from hepatocellular carcinoma using machine learning.

Sarthak Vajpayee[a,*], and Akansha Mangal[b], and Akansha Mangal[1]

[a]*Freelance Data Scientist.*
[b]*Department of Computer Science, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India.*

Abstract

Liver cancer is one of the most recurrent detected cancers in the world. The most common type of liver cancer is hepatocellular carcinoma (HCC) which begins in the cells called hepatocytes. It can be cured with surgery or transplant if detected early but is incurable in more advanced cases. The exact cause of HCC is unknown, but some factors like several demographic, risk factors, laboratory features, and underlying problems like hepatitis B and hepatitis C virus, autoimmune hepatitis, and heavy drinking increase the risk of death from HCC. In this paper, we have used some of these factors to predict their chances of survival of a patient diagnosed with HCC. For our study, we have used a publicly available dataset of 165 patients maintained and made available by a University Hospital in Portugal. The dataset contains 23 quantitative and 26 qualitative variables with 10.22% missing data. In our approach, we first standardized the data and then handled the missing values by comparing four imputation techniques: mean, median, KNN, and random-forest-based MICE. After imputation, ANOVA F-value and mutual information were used to select the relevant features. The prepared data was then studied on five classifiers: Logistic Regression, Support Vector Machine, Random-Forest, Bagging-Classifier, and Multilayer Perceptron (MLP). The MLP based classifier and random-forest-based MICE imputation technique and feature selection produced the best results with an accuracy of 90.02% and f1-score of 0.914. With these results, we can say that our machine-learning based approach can be used to test the performance with huge database and effectively be applied to predict mortality risk in patients and aid clinicians.

Keywords: `Machine learning, missing data imputation, feature selection, classification, hepatocellular carcinoma.`

1. Introduction

Hepatocellular carcinoma was the third most common cause of cancer deaths in 2020, with an estimated 905,677 new cases and 830,180 deaths [1]. Some of the risk factors like alcohol addiction, chronic hepatitis B and hepatitis C virus infection, metabolic liver disease, and exposure to dietary toxins are preventable to reduce the global issue of HCC [2]. Early-stage HCC is compliant to potentially curative treatments like surgical resection, local ablation, and liver transplantation [3].

Prevalence of HCC is rising in people born between 1945 and 1965 in the United States due to their advancing age and prolonged HCV infection [4]. According to the study of Ashish Kumar et al. [5], HCC is one of the major causes of mortality, morbidity and healthcare expenditure for patients with chronic liver disease in India. Over the last few decades, various techniques have been proposed by researchers for the detection of liver cancer, for example, using Otsu's method for enhancing MRI images and using watershed method for segmenting cancer cells in the processed image [6]. Amita Das et al. [7] proposed a new technique called watershed Gaussian based deep learning (WGDL) using 225 tomography (CT) images of the liver and used deep neural networks for the automated classification of different cancers. Xin Dong et al. [8] presented the hybridized fully convolutional neural network (HFCNN) method for identification and segmentation of liver cancer and lesions. N. Ramkumar et al. [9] used Conditional probability Bayes theorem for the prediction of liver cancer. To predict the recurrence of liver cancer, Hiroyuki Ogihara et al. [10] proposed a classifier based on Boolean algebra using a binary pattern consisting of a combination of clinical and genomic data. Some of the research studies have explored ways and suggestions to treat and prevent HCC, for example, one study reviews the essential workup and treatment options necessary for the treatment of patients with primary liver cancers [11]. Another study has provided an update on the current and future medical and surgical management of HCC [12]. In their study, Tracey G.Simon et al. have discussed various lifestyle and environmental approaches for the prevention of HCC [13]. According to Junhao Zheng et al., salvage liver transplantation and repeat hepatectomy might be the best treatment on the benefits of survival [14]. Many studies have also applied machine learning on different HCC datasets over the last years, for example, Santos et

*Corresponding author. Tel: +91-9919191103; Fax: n/a
Email address: `itssarthakvajpayee@gmail.com` (Sarthak Vajpayee)

al. proposed a cluster-based oversampling algorithm and used SMOTE for balancing the dataset [15]. This balanced data was used for the classification with logistic regression and neural network algorithms [16]. Sawhney et al. employed a random forest algorithm and firefly optimization method [17] for the diagnosis of HCC [18]. Ksiazek et al. employed a 2-level genetic optimizer for feature selection and parameter optimization and used naive Bayes, random forest, KNN, SVM, logistic regression, linear discriminant and, multilayer perceptron for the classification of the HCC dataset [19]. Fahrettin Burak Demir et al. employed mean imputation method for missing values and used chaotic darcy optimization method for feature selection and studied nine widely used machine learning based classifiers to classify the HCC dataset [20].

One of the major contributions of our work was to employ an efficient data preprocessing technique on the HCC dataset that aids classification models to yield better results. The most challenging part in data preprocessing was handling the missing values. Although there are only 10.22% missing values in the dataset, they are scattered in such a way that applying common missing data handling techniques like row and column sampling would lead to over 90% data loss and make the classification task quite challenging. We tested several well-known imputation techniques like mean, median, KNN and random forest based MICE imputation and concluded that the random forest based MICE imputation outperforms the other imputation techniques on this dataset. Other contributions include selecting the best features that are most relevant for the classification task. The dataset contains 50 features including the output label which represents whether a diagnosed patient was HCC positive or not. Most of the remaining 49 features were irrelevant and did not contribute to better accuracy. We used ANOVA F-value and mutual information for selecting the most relevant features for classification. For testing the preprocessed data, we compared the performances of five classification models – logistic regression based classifier, support vector machine classifier, random forest, bagged classifier and, neural networks – and concluded that neural networks outperformed the other classifiers by achieving an accuracy of 90.02% and f1-score of 0.914 on dataset preprocessed using random forest based MICE imputation and feature selection.

2. Data description

The HCC dataset was made available by UCI machine learning Repository [21], and contains several demographic, risk factors, laboratory and overall survival features of 165 real patients diagnosed with HCC at a University Hospital in Portugal. The dataset contains 50 features out of which one is the target label encoded as a binary variable where the value 1 denotes that the patient died and 0 denotes that the patient survived. The remaining 49 features were selected according to the management of HCC's EASL-EORTC (European Association for the Study of the Liver - European Organisation for Research and Treatment of Cancer) clinical practice guidelines [22]. Out of these 49 features, there are 23 categorical features, and 26 numeric features. The list of features provided in the dataset are listed in Table 1. In the dataset, 10.22% of the data contributes to missing values and is represented as '?'. A mild level of data imbalance is also present in the dataset as 63 patents have the target label 'deceased' and 102 patients have the target label 'survived'.

S. No	Prognostic features	Range	Missing (%)
1	Gender	0,1	0.0
2	Symptoms	0,1	10.91
3	Alcohol	0,1	0.0
4	Hepatitis B Surface Antigen	0,1	10.3
5	Hepatitis B e Antigen	0,1	23.64
6	Hepatitis B Core Antibody	0,1	14.55
7	Hepatitis C Virus Antibody	0,1	5.45
8	Cirrhosis	0,1	0.0
9	Endemic Countries	0,1	23.64
10	Smoking	0,1	24.85
11	Diabetes	0,1	1.82
12	Obesity	0,1	6.06
13	Hemochromatosis	0,1	13.94
14	Arterial Hypertension	0,1	1.82
15	Chronic Renal Insufficiency	0,1	1.21
16	Human Immunodeficiency Virus	0,1	8.48
17	Nonalcoholic Steatohepatitis	0,1	13.33
18	Esophageal Varices	0,1	31.52
19	Splenomegaly	0,1	9.09
20	Portal Hypertension	0,1	6.67
21	Portal Vein Thrombosis	0,1	1.82
22	Liver Metastasis	0,1	2.42
23	Radiological Hallmark	0,1	1.21
24	Age at diagnosis	20.0-93.0	0.0
25	Grams of Alcohol per day	0.0-500.0	29.09
26	Packs of cigarettes per year	0.0-510.0	32.12
27	Performance Status	0.0-4.0	0.0
28	Encephalopathy degree	1.0-3.0	0.61
29	Ascites degree	1.0-3.0	1.21
30	International Normalized Ratio	0.84-4.82	2.42
31	Alpha-Fetoprotein (ng/mL)	1.2-1810346.0	4.85
32	Hemoglobin (g/dL)	5.0-18.7	1.82
33	Mean Corpuscular Volume (Fl)	69.5-119.6	1.82
34	Leukocytes(G/L)	2.2-13000.0	1.82
35	Platelets (G/L)	1.71-459000.0	1.82
36	Albumin (mg/dL)	1.9-4.9	3.64
37	Total Bilirubin(mg/dL)	0.3-40.5	3.03
38	Alanine transaminase (U/L)	11.0-420.0	2.42
39	Aspartate transaminase (U/L)	17.0-553.0	1.82
40	Gamma glutamyl transferase (U/L)	23.0-1575.0	1.82
41	Alkaline phosphatase (U/L)	1.28-980.0	1.82
42	Total Proteins (g/dL)	3.9-102.0	6.67
43	Creatinine (mg/dL)	0.2-7.6	4.24
44	Number of Nodules	0.0-5.0	1.21
45	Major dimension of nodule (cm)	1.5-22.0	12.12
46	Direct Bilirubin (mg/dL)	0.1-29.3	26.67
47	Iron (mcg/dL)	0-224	47.88
48	Oxygen Saturation (%)	0-126	48.48
49	Ferritin (ng/mL)	0-2230	48.48
50	Class	0.0-1.0	0.0

Table 1: List of features provided in the HCC dataset along with their range and percentage of missing values.

3. Data Standardization

Data standardization is an important pre-processing step when the features have different ranges. It assures that the values of numeric features in the dataset are on a common scale and therefore comparable [23]. All the 26 numerical features were standardized to follow normal distribution with mean of zero and a standard deviation of one using Equation 1. The categorical features need not be standardized.

$$\mathbf{x}_{standardized} = \frac{\mathbf{x} - \mu}{\sigma} \tag{1}$$

Here, $\mathbf{x}$ represents the raw numerical feature vector, μ denotes the mean of feature vector $\mathbf{x}$ and, σ represents the standard deviation of feature vector $\mathbf{x}$. The process was applied on the dataset prior to the imputation of missing values. In Figure 1, 2, 3, three numerical features are shown before and after standardization (left to right). It is observable that features before standardization have different ranges which were improved after standardization.

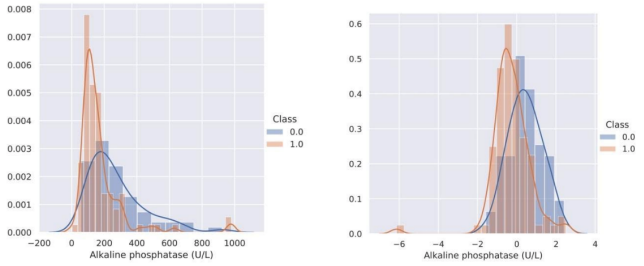

Figure 1: For the feature Alkaline phosphatase (U/L), the original distribution is represented on the left and the standardized distribution is represented on the right for both the target classes.

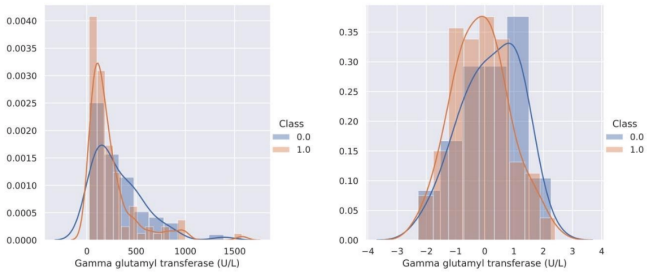

Figure 2: For the feature Gamma glutamyl-transferase (U/L), the original distribution is represented on the left and the standardized distribution is represented on the right for both the target classes.

4. Missing Values Imputation

For handling the missing values in the dataset we compared four imputation techniques, namely:

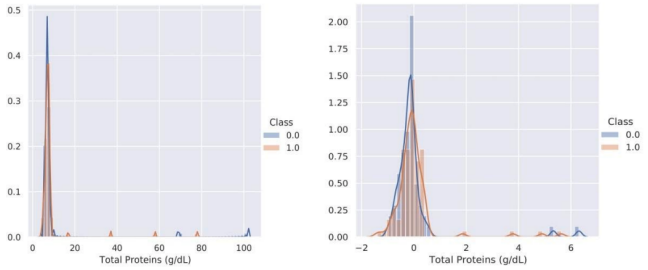

Figure 3: For the Total proteins (g/dL), the original distribution is represented on the left and the standardized distribution is represented on the right for both the target classes.

- Mean imputation
- Median imputation
- KNN imputation
- Random forest based Multiple Imputation by Chained Equations (MICE)

The process begins by first imputing the missing values in train sets and then using the imputed train sets to impute the missing values in test sets. This setup ensures that the data in test sets remains unseen and no missing values in the train data were imputed through test data since it would lead to data leakage. We used k-fold cross validation to split the data into multiple folds of train and test sets, as discussed in Subsection 8.1. The motive behind dividing the dataset into multiple fold was to utilize the complete dataset for a rigorous testing of the imputation and classification techniques. The four imputation techniques mentioned before are briefly described below:

4.1. Mean Value Imputation

Mean value imputation is a well known imputation technique applied to handle the missing values in a dataset [24]. This imputation process begins by iterating over the features in the train dataset one by one and groups them into two sets based on their target label, which in our case is binary encoded. Set one contains data points that have the target label 0 and set two contains data points that have the target label 1. The missing values in each set are replaced by the mean of the non-missing values in that set. Once the process is complete, all the missing values in the train dataset are imputed and utilized to impute the missing values in the test dataset. For each feature in the test dataset, the missing values in the feature are replaced by the mean of all the values in that feature in the train dataset.

4.2. Median Value Imputation

Median value imputation is another well known imputation technique applied to handle the missing values in a dataset. It

is similar to the mean value imputation technique since it uses the median statistic instead of the mean statistic in the process and therefore it has been used in many works alongside mean imputation for comparison [25], [26]. This imputation process begins by iterating over the features in train dataset one by one and groups them into two sets based on their target label. In our case, set one contains data points that have the target label 0 and set two contains data points that have the target label 1. The missing values in each set are replaced by the median of the non-missing values in the set. Once the process is complete, all the missing values in the train dataset are imputed and utilized to impute the missing values in the test dataset. For each feature in the test dataset, the missing values in the feature are replaced by the median of all the values in that feature in the train dataset.

4.3. KNN Imputation

The k nearest neighbour (KNN) imputation technique utilizes the data-points near the missing value to calculate the imputation value. These nearby data points are calculated by means of distance metrics like euclidean distance, manhattan distance, cosine distance etc. The algorithm begins by calculating the distance between the missing datapoint and the other data points in train data. It then selects k data points nearest to the missing value datapoint and computes the missing value by taking the mean of non-missing values from the k neighbours. The algorithm has been used in a variety of research works like missing value imputation for DNA microarray data [27] and imputation for software quality datasets [28]. The variable k is a hyperparameter and in our work the value of k was decided experimentally as 13.

4.4. Multiple Imputation by Chained Equations (MICE) with random forest

Multivariate imputation by chained equations (MICE) is one of the most favourable algorithm for handling the missing values and is available in many programming languages like R due to it's popularity [29], [30]. Random forest based MICE is a variation of the MICE imputation technique [31] and has been widely used in many studies for handling missing data [32], [33]. The algorithm begins by creating multiple copies of the dataset and replacing the missing data values in each copy using the MICE procedure. It then analyzes the copies of the dataset using an intended statistical analysis and combines the results of these data analyses and returns it.

The MICE procedure imputes the missing data by cycling multiple times through the steps below:

Step 1: The missing values in each feature are replaced with a temporary "placeholder" value using the median of non-missing values available for that feature.

Step 2: The "placeholder" values in the current feature are set back to missing.

Step 3: The missing values of the current feature are predicted using a random forest model trained on the remaining features.

Step 4: Steps 2–3 are repeated separately for each feature that has missing data.

Steps 1 - 4 are repeated once for each of the features. At the end of this cycle, all of the missing values in the features are replaced with predictions from the trained random forest models.

The above procedure was done on multiple copies of the original data after which, the imputed copies of the dataset are combined to create the final imputed dataset. The number of copies is a hyperparameter which in our study was decided experimentally as 5.

5. Feature selection

Feature selection is an important step that insures only the relevant features are selected from the dataset for training the classification models [34]. The primary objectives of feature selection are:

(a) Choosing the best feature subset in order to minimize the generalization error.

(b) Improve the generalization performance.

(c) Fasten the process by reducing the computational cost.

In this work, we have used ANOVA F-value method for numerical features and the mutual information method for categorical features. The two methods are briefly described below:

5.1. ANOVA F-value:

Analysis of variance (ANOVA) is a statistical method that uses the F-statistic hypothesis test or F-test to determine whether the mean of groups of three or more distributions are similar or not by comparing intra-group variance to inter-group variance. Important features are selected by considering the difference in mean of sample distributions such that, greater difference in mean denotes that the feature is more important. Due to it's promising results, this technique has been used in many studies like pattern recognition of jet fuels [35], Identification of bacteriophage virion proteins [36] and, Modular neural-SVM scheme for speech emotion recognition [37]. The process has a hyperparameter n which denotes the number of top important features to select. In this study, value of n was decided experimentally as 11.

5.2. Mutual information:

Mutual information is a technique that determines the amount of information obtained from one variable given the known value of other variable [38]. The mutual information between two variables can be calculated using Equation 2.

$$I(X;Y) = H(X) - H(X|Y) \tag{2}$$

Here, I(X;Y) is the mutual information for X and Y represented in bits, H(X) is the entropy for X and H(X—Y) is the conditional entropy for X given Y. The process has a hyperparameter n which denotes the number of top important features to select. In this study, value of n was decided experimentally as 20.

6. Classification

In this study we have trained and tested five different models on the HCC dataset to evaluate the four imputation techniques and selected important features. The five classifiers are based on logistic regression, support vector machine (SVM), random forest, bagging classifier and, multilayer perceptron (MLP) and are briefly described below:

6.1. Logistic Regression:

Logistic regression is a modeling technique that uses logistic function to model a dependent binary variable and deduce the relationship between the dependent binary variable and one or more interval, ordinal, or nominal variables [39]. The logistic regression model has shown promising results in various tasks like bankruptcy prediction from user data [40] and, landslide susceptibility mapping in the Kakuda-Yahiko Mountains, Central Japan [41]. The output y_{pred} of the logistic regression is given by Equations 3 and 4.

$$y_{pred} = \sigma(\mathbf{w}^T.\mathbf{x} + b) \tag{3}$$

$$\sigma(z) = \frac{1}{1 + (e^{-z})} \tag{4}$$

Here, $\mathbf{w}$ and b are the trainable parameters known as weight vector and intercept term respectively, $\mathbf{x}$ is the input vector and σ is the logistic or sigmoid function. To train this model, we used mini-batch gradient descent optimizer along with $l1$ penalty (lasso) and $l2$ penalty (ridge). The hyperparameters tested during optimizing the classifier are listed in Table 2. The best parameters were selected using k-fold cross validation as discussed in Subsection 8.1.

6.2. Support Vector Machine:

Support Vector Machine is a classification strategy that works by transforming the training data into a higher dimension, and using it to calculate the optimal separation boundaries between classes [42]. In SVMs, these separation boundaries are referred to as hyper planes and are identified using support vectors. Using these separation boundaries, SVMs are able to classify both linear and nonlinear data efficiently. This is also a popular classifier and has been successfully applied in the field of bioinformatics [43], hydrology [44], and for financial forecasting [45].

$$\sigma(\mathbf{w}) = \frac{1}{2}\mathbf{w^T w} - \mathbf{J}(\mathbf{w}, b, a) \tag{5}$$

Here, $\mathbf{w}$ and b are the trainable parameters and $\mathbf{J}(\mathbf{w}, b, a)$ is the loss function to be minimized. K-fold cross validation was used for optimizing the classifier hyperparameters as discussed in Subsection 8.1 and the hyperparameters tested for optimizing the classifier are listed in Table 2.

6.3. Random Forest:

Random forest is an ensemble learning technique which consists of multiple decision trees as base learners, and uses bootstrap aggregation to create random samples of features from the input dataset and train each individual tree. The uncorrelated forest of trees created is then used to predict the output value by majority or committee voting [46]. Random forest is different from other bagging techniques in terms of feature sampling as it uses only a subset of features at random and the best split feature from the subset is used to split each node in a tree. Because of it's effectiveness, random forest classifiers have been successfully applied in tasks like modeling structure-activity relationships of pharmaceutical molecules [47] and, groundwater potential mapping in Mehran Region, Iran [48]. Several hyperparameters that were testing during optimization of the algorithm are listed in Table 2.

6.4. Bagging Classifier:

Bagging classifier is another ensemble learning technique that fits the base classifiers (decision trees) on random subsets of the input dataset and then utilizes their individual predictions to deduce the final prediction. This method reduces the variance of a single decision tree, by introducing randomization into its construction procedure [49]. Unlike random forests, bagging classifier uses only row sampling, instead of both row and column sampling. Several hyperparameters that were testing during optimization of the algorithm are listed in Table 2.

6.5. Multilayer perceptron:

Multilayer perceptron is a type of artificial neural network (ANN) with multiple hidden layers between the input and output layers. this models is very efficient in capturing complex nonlinear behaviours in data due to its flexible structure and approximation capabilities [50]. The architecture of a MLP comprises a collection of connected units called neurons that are organized into multiple layers such that the neurons of one layer are connected only to neurons of immediately preceding and immediately following layers. This neural network based classifier is quiet popular for it's efficacy and has wide range of applications in tasks like damage detection of truss bridge joints [51] and, breast cancer detection [52]. Each node in MLP classifier calculates the output from the input data using Equation 6.

$$\mathbf{a}_j^l = \phi(b_j^l + \sum_k \mathbf{W}_{jk}^l \mathbf{a}_k^{l-1}) \tag{6}$$

Where $\mathbf{W}_{jk}^l$ is the trainable weight matrix of current layer l, $\mathbf{a}_k^{l-1}$ is the activation vector from the previous layer $l-1$, b_j^l is the bias term of current layer and, ϕ is the non-linear activation function which in our experiments was softmax activation function for the output layer and ReLU activation function for the remaining layers.

The model uses supervised learning to update the trainable parameters through backpropagation. For training the model, adaptive momentum (ADAM) was used as the optimization function and sparse categorical cross-entropy was used as the loss function. Other hyperparameters that were tested for optimizing the classifier are listed in Table 2.

7. Evaluation metrics

For comparing different model architecture and assessing their performances, we chose F1-Score and accuracy as the evaluation metrics. The two evaluation metrics are briefly described below.

7.1. Accuracy:

Accuracy is a statistical measure used for evaluating the performance of a classification model. It is defined as a ratio of the number of correctly classified points to the total number of points and is usually denoted in percentage. The accuracy value of a model can be derived from a confusion matrix (illustrated in Figure 4) using Equation 7.

$$\begin{aligned} Accuracy &= \frac{\text{Number of correctly classified points}}{\text{Total number of classified points}} \\ &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \end{aligned} \tag{7}$$

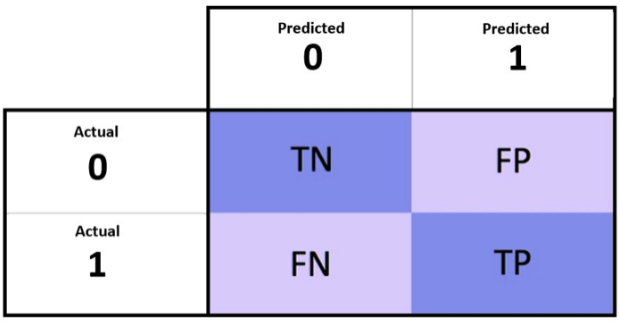

Figure 4: Confusion Matrix represented in terms of actual class labels and predicted class labels.

Here, true positives (TP) is the number of positive data points that were correctly classified as positives, true negatives (TN) is the number of negative data points that were correctly classified as negatives, false positives (FP) is the number of negative data points that were incorrectly classified as positives, and false negative (FN) is the number of positive data points that were incorrectly classified as negatives.

7.2. F1-Score:

F1-Score is another statistical measure used for evaluating the performance of a classification model. It is defined as harmonic mean of precision and recall values, which are derived from a confusion matrix (illustrated in Figure 4) using Equations 8, 9, and 10.

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{10}$$

A high F1-Score is only possible when precision and recall values are both high which means that the model is effectively classifying the data resulting in a lesser number of false positives and false negatives.

8. Experimental setup and Results

For testing and evaluating the effectiveness of the different imputation techniques, feature selection methods, and selecting the best classification model, HCC dataset was first divided into two non-overlapping sets namely optimization set and test set. The optimization set was used for optimizing the hyperparameters using k-fold cross-validation, and the test set was later used to evaluate the performances of the optimized classifiers and data pre-processing techniques on unseen data. K-fold cross validation and evaluation metric scores on the test set are discussed in the subsections below.

Various elements of the resources that were used for conducting the experiments in this study include python programming language, along with Pandas, Tensorflow, Seaborn, Matplotlib and Scikit-learn libraries on an Intel Core i5-9600K 3.7 GHz machine with 16 GB of RAM.

8.1. K-fold cross validation:

K-fold cross validation is a statistical method based on a dataset resampling procedure to train, optimise and evaluate classification models [53]. In our experiments, we divided the dataset into k sets of non-overlapping datapoints such that one out of these k sets was chosen as validation set and the remaining $k-1$ sets were used as training set. The training and validation sets were used for optimizing the hyperparameters listed in Table 2. Since this is a k step process, the final performance was calculated by averaging the metric scores obtained in the k steps. Figure 5 illustrates k-fold cross validation for $k = 5$, as also used in our experiments and, Table 2 shows the list of hyperparameters that were tested and selected using k-fold cross validation.

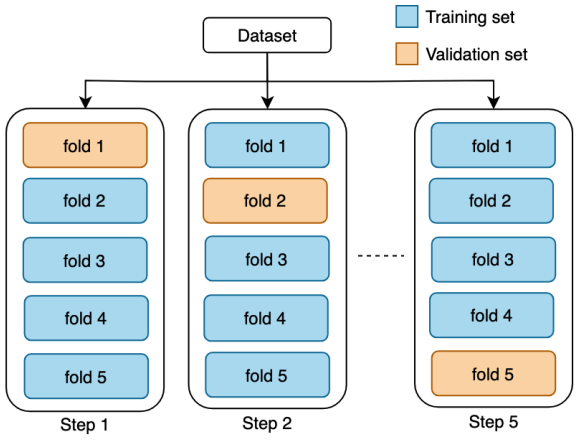

Figure 5: Five step data splitting for k-fold cross validation. At each step, training and validation set is used for hyperparameter optimization.

8.2. Results:

After obtaining the optimal hyperparameters through k-fold cross validation, the complete optimization set was preprocessed and used for training the five classifiers. The best preprocessing techniques and classifier were then selected based on the metric scores on the test set.

During feature selection, 20 categorical and 11 numerical features were selected using mutual information and ANOVA F-Measures respectively. The selected features are listed in Table 3.

Numerical features	**Categorical features**
Performance Status	Gender
Ascites degree	Symptoms
International Normalised Ratio	Alcohol
Alpha-Fetoprotein (ng/mL)	Hepatitis B Surface Antigen
Haemoglobin (g/dL)	Hepatitis B Core Antibody
Albumin (mg/dL)	Hepatitis C Virus Antibody
Total Bilirubin(mg/dL)	Cirrhosis
Aspartate transaminase (U/L)	Endemic Countries
Alkaline phosphatase (U/L)	Smoking
Direct Bilirubin (mg/dL)	Diabetes
Iron (mcg/dL)	Obesity
	Hemochromatosis
	Human Immunodeficiency Virus
	Nonalcoholic Steatohepatitis
	Esophageal Varices
	Splenomegaly
	Portal Hypertension
	Portal Vein Thrombosis
	Liver Metastasis
	Radiological Hallmark

Table 3: List of selected Categorical and Numerical features.

Performances of the classifiers on unseen data with and without feature selection are shown in Table 4 and Table 5 respectively. Metric results of the best performing classifier (MLP) over the different imputation techniques are illustrated in Figure 6 and Figure 7. It can be observed that in both cases, random forest based MICE imputation method performs better than the other imputation techniques. Also, multilayer-perceptron outperforms all the other classifiers by achieving an accuracy of 90.02% and f1-score of 0.91 on MICE imputed dataset with feature selection. It is also observable that feature selection plays a significant role for in improving the performances of the models on the HCC dataset and the imputation techniques perform better when feature selection is used.

Table 2: The hyperparameters tested while preparing the data and extracting important features using k-fold cross validation.

Classifier type	Hyperparameter	Tested values	Best value
Logistic regression	penalty	$l1, l2, elasticnet, none$	$l2$
	C (Regularization parameter)	10^{-k}, where $k = 0, 1, 2, 3, 4$	10^{-3}
SVM	C (Regularization parameter)	10^{-k}, where $k = 0, 1, 2, 3, 4$	10^{-3}
	kernel type	$linear, poly, rbf$	$poly$
	Degree of the polynomial kernel function	$1, 2, 3, 4$	3
Random forest	n_estimators (number of base learners)	$32 \times k$, where $k = 1, 2, 3, 4, 5, 8$	64
	max_depth	$2 \times k$, where $k = 1, 2, 3, 4, 5$	6
	min_samples_split	$2 \times k$, where $k = 1, 2, 3, 4, 5$	4
Bagging classifier	n_estimators (number of base learners)	2^k, where $k = 4, 5, 6, 7, 8, 9$	128
	max_samples	$k \times 10^{-1}$, where $k = 3, 4, 5, 6, 7, 8$	0.5
	max_features	$k \times 10^{-1}$, where $k = 3, 4, 5, 6, 7, 8$	0.5
MLP	Number of hidden layers	$1, 2, 3, 4, 5, 6$	3
	Number of units in each layer	2^k, where $k = 2, 3, 4, 5, 6, 7$	32
	Dropout rate	$0.2, 0.3, 0.4, 0.5, 0.6, 0.7$	0.4
	Optimization function	Adam, RMSprop, AdaGrad	Adam
	epochs	$20 \times k$, where $k = 1, 2, 3, 4, 5$	60
	learning rate	10^{-k}, where $k = 2, 3, 4, 5, 6$	10^{-5}
	batch-size	2^k, where $k = 4, 5, 6, 7$	32

Table 4: Performance evaluation of different imputation techniques with feature selection.

	Logistic regression		**SVM**		**Random forest**		**Bagging classifier**		**MLP**	
	Accuracy (%)	**F1-score**	**Accuracy (%)**	**F1-score**	**Accuracy (%)**	**F1-score**	**Accuracy (%)**	**F1-score**	**Accuracy (%)**	**F1-score**
Mean	77.29	0.805	56.08	0.715	56.08	0.693	74.86	0.815	76.68	0.811
Median	74.26	0.794	62.14	0.754	62.14	0.545	74.86	0.809	78.89	0.825
KNN	65.17	0.716	65.17	0.716	50.02	0.585	73.65	0.785	81.53	0.841
MICE(RF)	**71.23**	**0.785**	**73.05**	**0.798**	**76.08**	**0.816**	**76.68**	**0.812**	**90.02**	**0.914**

Table 5: Performance evaluation of different imputation techniques without feature selection.

	Logistic regression		**SVM**		**Random forest**		**Bagging classifier**		**MLP**	
	Accuracy (%)	**F1-score**	**Accuracy (%)**	**F1-score**	**Accuracy (%)**	**F1-score**	**Accuracy (%)**	**F1-score**	**Accuracy (%)**	**F1-score**
Mean	62.13	0.685	46.98	0.607	46.98	0.549	73.08	0.813	75.47	0.79
Median	59.1	0.719	56.07	0.7	43.95	0.48	73.65	0.79	76.89	0.822
KNN	59.1	0.73	56.07	0.715	43.95	0.166	73.05	0.764	80.53	0.834
MICE(RF)	**70.01**	**0.765**	**72.43**	**0.782**	**76.68**	**0.808**	**76.08**	**0.793**	**88.8**	**0.902**

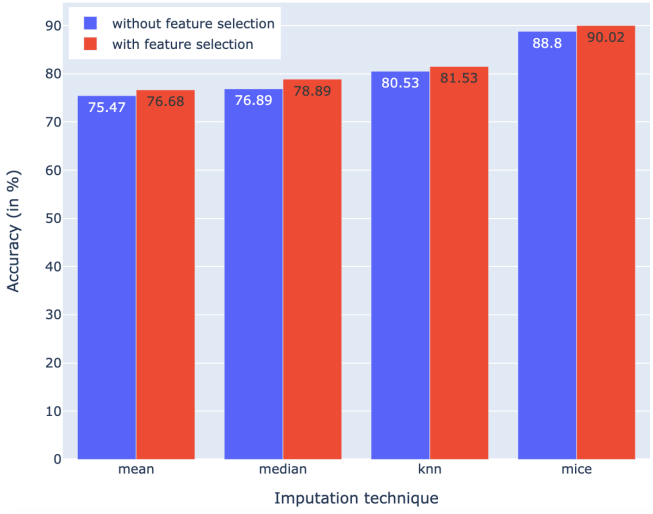

Figure 6: Bar graph of accuracy on MLP with and without feature selection.

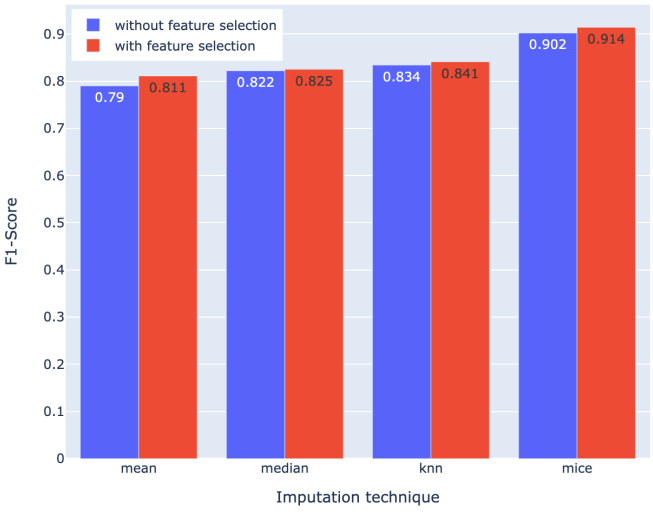

Figure 7: Bar graph of f1-score on MLP with and without feature selection

9. Conclusion

In this study, we have compared four imputation techniques, namely, mean imputation, median imputation, KNN imputation and random forest based MICE imputation, for handling the missing values along with feature selection in the HCC dataset. K-fold cross validation was used for data preprocessing and optimizing the related hyperparameters. The imputation and feature selection techniques were compared based on the performances of the five selected classification models on the preprocessed dataset. The results show that random forest based MICE imputation along with feature selection outperforms the other imputation techniques by achieving an accuracy of 90.02% and f1-score of 0.91 using MLP classifier. The results indicate that this machine-learning based approach can be effectively used for predicting mortality risk in patients suffering from HCC.

10. Future work

Although the results are promising, the setup requires more data for rigorously testing the robustness of the pre-processing techniques and classification models. To address the issue, data using multiple other HCC datasets are being collected and used for enhancing the approach.

References

[1] L. Cicalese, B. Cagir, What is the global incidence of hepatocellular carcinoma (hcc) worldwide? (Feb 2021).

[2] P. Rawla, T. Sunkara, P. Muralidharan, J. P. Raj, Update in global trends and aetiology of hepatocellular carcinoma, Contemp Oncol (Pozn) 22 (3) (2018) 141–150.

[3] J. D. Yang, P. Hainaut, G. J. Gores, A. Amadou, A. Plymoth, L. R. Roberts, A global view of hepatocellular carcinoma: trends, risk, prevention and management, Nat Rev Gastroenterol Hepatol 16 (10) (2019) 589–604.

[4] C. Gadiparthi, E. R. Yoo, V. S. Are, P. Charilaou, D. Kim, G. Cholankeril, C. Pitchumoni, A. Ahmed, Hepatocellular carcinoma is leading in cancer-related disease burden among hospitalized baby boomers, Ann Hepatol 18 (5) (2019) 679–684.

[5] A. Kumar, S. K. Acharya, S. P. Singh, A. Arora, R. K. Dhiman, R. Aggarwal, A. C. Anand, P. Bhangui, Y. K. Chawla, S. D. Gupta, V. K. Dixit, A. Duseja, N. Kalra, P. Kar, S. S. Kulkarni, R. Kumar, M. Kumar, R. Madhavan, V. M. Prasad, A. Mukund, A. Nagral, D. Panda, S. B. Paul, P. N. Rao, M. Rela, M. K. Sahu, V. A. Saraswat, S. R. Shah, Shalimar, P. Sharma, S. Taneja, M. Wadhawan, 2019 update of indian national association for study of the liver consensus on prevention, diagnosis, and management of hepatocellular carcinoma in india: The puri II recommendations, Journal of Clinical and Experimental Hepatology 10 (1) (2020) 43–80. doi:10.1016/j.jceh.2019.09.007.
URL https://doi.org/10.1016/j.jceh.2019.09.007

[6] A. Dutta, A. Dubey, Detection of liver cancer using image processing techniques, in: 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0315–0318. doi:10.1109/ICCSP.2019.8698033.

[7] A. Das, U. R. Acharya, S. S. Panda, S. Sabut, Deep learning based liver cancer detection using watershed transform and gaussian mixture model techniques, Cognitive Systems Research 54 (2019) 165–175. doi:

https://doi.org/10.1016/j.cogsys.2018.12.009.
URL https://www.sciencedirect.com/science/article/pii/S1389041718310143

[8] X. Dong, Y. Zhou, L. Wang, J. Peng, Y. Lou, Y. Fan, Liver cancer detection using hybridized fully convolutional neural network based on deep learning framework, IEEE Access 8 (2020) 129889–129898. doi:10.1109/ACCESS.2020.3006362.

[9] N. Ramkumar, S. Prakash, S. A. Kumar, K. Sangeetha, Prediction of liver cancer using conditional probability bayes theorem, in: 2017 International Conference on Computer Communication and Informatics (ICCCI), 2017, pp. 1–5. doi:10.1109/ICCCI.2017.8117752.

[10] H. Ogihara, Y. Fujita, Y. Hamamoto, N. Iizuka, M. Oka, Classification based on boolean algebra and its application to the prediction of recurrence of liver cancer, in: 2013 2nd IAPR Asian Conference on Pattern Recognition, 2013, pp. 838–841. doi:10.1109/ACPR.2013.152.

[11] J. C. Mejia, J. Pasko, Primary Liver Cancers: Intrahepatic Cholangiocarcinoma and Hepatocellular Carcinoma, Surg Clin North Am 100 (3) (2020) 535–549.

[12] S. Daher, M. Massarwa, A. A. Benson, T. Khoury, Current and Future Treatment of Hepatocellular Carcinoma: An Updated Comprehensive Review, J Clin Transl Hepatol 6 (1) (2018) 69–78.

[13] T. G. Simon, A. T. Chan, Lifestyle and Environmental Approaches for the Primary Prevention of Hepatocellular Carcinoma, Clin Liver Dis 24 (4) (2020) 549–576.

[14] J. Zheng, J. Cai, L. Tao, M. A. Kirih, Z. Shen, J. Xu, X. Liang, Comparison on the efficacy and prognosis of different strategies for intrahepatic recurrent hepatocellular carcinoma: A systematic review and Bayesian network meta-analysis, Int J Surg 83 (2020) 196–204.

[15] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, Smote: Synthetic minority over-sampling technique, J. Artif. Intell. Res. (JAIR) 16 (2002) 321–357. doi:10.1613/jair.953.

[16] M. S. Santos, P. H. Abreu, P. J. García-Laencina, A. Simão, A. Carvalho, A new cluster-based oversampling method for improving survival prediction of hepatocellular carcinoma patients, J Biomed Inform 58 (2015) 49–59.

[17] N. Johari, A. Zain, N. Mustaffa, A. Udin, Firefly algorithm for optimization problem, Applied Mechanics and Materials 421 (04 2013). doi:10.4028/www.scientific.net/AMM.421.512.

[18] R. Sawhney, P. Mathur, R. Shankar, A Firefly Algorithm Based Wrapper-Penalty Feature Selection Method for Cancer Diagnosis, 2018, pp. 438–449. doi:10.1007/978-3-319-95162-1_30.

[19] W. Ksiazek, M. Abdar, U. R. Acharya, P. Plawiak, A novel machine learning approach for early detection of hepatocellular carcinoma patients, Cognitive Systems Research 54 (2019) 116–127. doi:https://doi.org/10.1016/j.cogsys.2018.12.001.
URL https://www.sciencedirect.com/science/article/pii/S1389041718308714

[20] F. Demir, T. Tuncer, A. Kocamaz, F. Ertam, A survival classification method for hepatocellular carcinoma patients with chaotic darcy optimization method based feature selection, Medical Hypotheses 139 (2020) 109626. doi:10.1016/j.mehy.2020.109626.

[21] D. Dua, C. Graff, UCI machine learning repository (2017).
URL http://archive.ics.uci.edu/ml

[22] J. M. Llovet, M. Ducreux, R. Lencioni, A. M. Di Bisceglie, P. R. Galle, J. F. Dufour, T. F. Greten, E. Raymond, T. Roskams, T. De Baere, M. Ducreux, V. Mazzaferro, M. Bernardi, EASL-EORTC clinical practice guidelines: management of hepatocellular carcinoma, J Hepatol 56 (4) (2012) 908–943.

[23] M. Gal, D. Rubinfeld, Data standardization, SSRN Electronic Journal (01 2018). doi:10.2139/ssrn.3326377.

[24] D. M. G, B. L. Bharadwaj, S. Vardhan, C. Gogulamudi, A Normalized Mean Algorithm for Imputation of Missing Data Values in Medical Databases, 2020, pp. 773–781. doi:10.1007/978-981-15-3172-9_72.

[25] S. Khan, A. Latiful Haque, Sice: an improved missing data imputation technique, Journal of Big Data 7 (06 2020). doi:10.1186/s40537-020-00313-w.

[26] X. Zhou, G. Eckert, W. Tierney, Multiple imputation in public health research, Statistics in medicine 20 (2001) 1541–9. doi:10.1002/sim.689.

[27] P. Keerin, W. Kurutach, T. Boongoen, Cluster-based knn missing value imputation for dna microarray data, 2012, pp. 445–450. doi:10.1109/ICSMC.2012.6377764.

[28] J. Huang, J. W. Keung, F. Sarro, Y.-F. Li, Y. Yu, W. Chan, H. Sun, Cross-validation based k nearest neighbor imputation for software quality datasets: An empirical study, Journal of Systems and Software 132 (2017) 226–252. doi:https://doi.org/10.1016/j.jss.2017.07.012.
URL https://www.sciencedirect.com/science/article/pii/S0164121217301516

[29] M. Azur, E. Stuart, C. Frangakis, P. Leaf, Multiple imputation by chained equations: What is it and how does it work?, International journal of methods in psychiatric research 20 (2011) 40–9. doi:10.1002/mpr.329.

[30] S. Buuren, C. Groothuis-Oudshoorn, Mice: Multivariate imputation by chained equations in r, Journal of Statistical Software 45 (12 2011). doi:10.18637/jss.v045.i03.

[31] F. Tang, H. Ishwaran, Random forest missing data algorithms (01 2017).

[32] A. K. Waljee, A. Mukherjee, A. G. Singal, Y. Zhang, J. Warren, U. Balis, J. Marrero, J. Zhu, P. D. Higgins, Comparison of imputation methods for missing laboratory data in medicine, BMJ Open 3 (8) (2013). arXiv:https://bmjopen.bmj.com/content/3/8/e002847.full.pdf, doi:10.1136/bmjopen-2013-002847.
URL https://bmjopen.bmj.com/content/3/8/e002847

[33] A. D. Shah, J. W. Bartlett, J. Carpenter, O. Nicholas, H. Hemingway, Comparison of Random Forest and Parametric Imputation Models for Imputing Missing Data Using MICE: A CALIBER Study, American Journal of Epidemiology 179 (6) (2014) 764–774. arXiv:https://academic.oup.com/aje/article-pdf/179/6/764/17341607/kwt312.pdf, doi:10.1093/aje/kwt312.
URL https://doi.org/10.1093/aje/kwt312

[34] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. 50 (6) (Dec. 2017). doi:10.1145/3136625.
URL https://doi.org/10.1145/3136625

[35] K. J. Johnson, R. E. Synovec, Pattern recognition of jet fuels: comprehensive gc× gc with anova-based feature selection and principal component analysis, Chemometrics and Intelligent Laboratory Systems 60 (1-2) (2002) 225–237.

[36] H. Ding, P.-M. Feng, W. Chen, H. Lin, Identification of bacteriophage virion proteins by the anova feature selection and analysis, Molecular BioSystems 10 (8) (2014) 2229–2235.

[37] M. Sheikhan, M. Bejani, D. Gharavian, Modular neural-svm scheme for speech emotion recognition using anova feature selection method, Neural Computing and Applications 23 (1) (2013) 215–227.

[38] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Physical review E 69 (6) (2004) 066138.

[39] D. Pregibon, Logistic Regression Diagnostics, The Annals of Statistics 9 (4) (1981) 705 – 724. doi:10.1214/aos/1176345513.
URL https://doi.org/10.1214/aos/1176345513

[40] E. K. Laitinen, T. Laitinen, Bankruptcy prediction: Application of the taylor's expansion in logistic regression, International Review of Financial Analysis 9 (4) (2000) 327–349. doi:https://doi.org/10.1016/S1057-5219(00)00039-9.
URL https://www.sciencedirect.com/science/article/pii/S1057521900000399

[41] L. Ayalew, H. Yamagishi, The application of gis-based logistic regression for landslide susceptibility mapping in the kakuda-yahiko mountains, central japan, Geomorphology 65 (1) (2005) 15–31. doi:https://doi.org/10.1016/j.geomorph.2004.06.010.
URL https://www.sciencedirect.com/science/article/pii/S0169555X04001631

[42] W. S. Noble, What is a support vector machine?, Nature Biotechnology 24 (12) (2006) 1565–1567. doi:10.1038/nbt1206-1565.
URL https://doi.org/10.1038/nbt1206-1565

[43] E. Byvatov, G. Schneider, Support vector machine applications in bioinformatics, Applied bioinformatics 2 (2) (2003) 67—77.
URL http://europepmc.org/abstract/MED/15130823

[44] S. Raghavendra. N, P. C. Deka, Support vector machine applications in the field of hydrology: A review, Applied Soft Computing 19 (2014) 372–386. doi:https://doi.org/10.1016/j.asoc.2014.02.002.
URL https://www.sciencedirect.com/science/article/pii/S1568494614000611

[45] T. Trafalis, H. Ince, Support vector machine for regression and applications to financial forecasting, in: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Vol. 6, 2000, pp. 348–353 vol.6. doi:10.1109/IJCNN.2000.859420.

[46] G. Biau, E. Scornet, A random forest guided tour, TEST 25 (2) (2016) 197–227. doi:10.1007/s11749-016-0481-7.
URL https://doi.org/10.1007/s11749-016-0481-7

[47] V. Svetnik, A. Liaw, C. Tong, T. Wang, Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules, Vol. 3077, 2004, pp. 334–343. doi:10.1007/978-3-540-25966-4_33.

[48] O. Rahmati, H. R. Pourghasemi, A. Melesse, Application of gis-based data driven random forest and maximum entropy models for groundwater potential mapping: A case study at mehran region, iran, Catena 137 (2015) 360–372. doi:10.1016/j.catena.2015.10.010.

[49] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140. doi:10.1007/BF00058655.
URL https://doi.org/10.1007/BF00058655

[50] P. Marius, V. Balas, L. Perescu-Popescu, N. Mastorakis, Multilayer perceptron and neural networks, WSEAS Transactions on Circuits and Systems 8 (07 2009).

[51] M. Mehrjoo, N. Khaji, H. Moharrami, A. Bahreininejа, Damage detection of truss bridge joints using artificial neural networks, Expert Systems with Applications 35 (2008) 1122–1131. doi:10.1016/j.eswa.2007.08.008.

[52] A. F. M. Agarap, On breast cancer detection: An application of machine learning algorithms on the wisconsin diagnostic dataset, in: Proceedings of the 2nd International Conference on Machine Learning and Soft Computing, ICMLSC '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 5–9. doi:10.1145/3184066.3184080.
URL https://doi.org/10.1145/3184066.3184080

[53] S. Raschka, Model evaluation, model selection, and algorithm selection in machine learning (2020). arXiv:1811.12808.