Development and Comparative Analysis of an Instance-Based Machine Learning Classifier

Govind Agarwal[a], Chirag Goel[b], Chunduru Sri Abhijit[c] and Alok Chauhan[d]
School of Computing Sciences, Vellore Institute of Technology, Chennai, Tamil Nadu, India

Keywords: Classification, Machine Learning, Lazy-Learner.

Abstract: Classification algorithms make it easy to classify many real-world problems, but they come with some cost. The existing classification algorithms have complex architectures, which can sometimes make the classification task tedious. This paper introduces a classification algorithm, which aims to improve upon existing methods by incorporating class count as a target feature. In this study, we attempt to offer a classification method that works with three different categories of datasets, viz., categorical, numerical, and a mixture of categorical and numerical. Firstly, for each input feature attribute, proposed algorithm counts the majority class of the target variable to train the model. Then it determines which class has appeared the most, after computing the majority class for each input characteristic. Final output of the classification algorithm would be the class that showed up the most. If there is a tie in the number of attributes, the class with the greater total count wins. Instance can belong to any class if the total count is also the same. Obviously, any attribute, which has the same count across all classes, is redundant or has no bearing on classification. This classification process is compared against several machine learning methods like KNN, logistic classifier and other models. Experimental results on various benchmark datasets demonstrate that the proposed algorithm is reliable and is promising with respect to several state-of-the-art classification methods in terms of classification accuracy as well as computational efficiency.

1 INTRODUCTION

Classification machine learning algorithms are a subset of supervised learning techniques, designed to discover a mapping between input data and output labels. Many fields have developed and adapted classification algorithms to tackle and automate a variety of practical problems (I.H. Sarker, 2021). Classification algorithms are defined here as a task of identifying the correct category of unseen data, based on the characteristics of previously seen classes (Tammy Jiang, 2020).

There are numerous classification algorithms, each having its advantages and disadvantages. Each algorithm utilizes a different approach to divide the data into classes, with some depending on simple, linear decision boundaries and others using more complicated, nonlinear ones. Many factors must be considered while deciding on an algorithm, such as the dataset's size and complexity, the type of characteristics to be used as inputs, and the level of precision and interpretability that must be achieved.

The type of classification developed during this research is a lazy learner. K-nearest neighbours and locality-sensitive hashing are two famous examples of classification methods for lazy learners. In addition to selecting an effective method, it is essential to pre-process and prepare the data before training the model to ensure that it is representative, balanced, and free of mistakes or outliers.

This research paper's primary objective is to look into following three variants for classification algorithm:

For datasets containing only categorical values

For datasets containing only numerical values

For datasets containing a mixture of categorical and numerical values.

[a] https://orcid.org/0009-0008-4673-8090
[b] https://orcid.org/0009-0009-8336-0987
[c] https://orcid.org/0009-0001-4108-2888
[d] https://orcid.org/0000-0002-8309-3403

Agarwal, G., Goel, C., Abhijit, C. and Chauhan, A.
Development and Comparative Analysis of an Instance-Based Machine Learning Classifier.
DOI: 10.5220/0012509800003739

In *Proceedings of the 1st International Conference on Artificial Intelligence for Internet of Things: Accelerating Innovation in Industry and Consumer Electronics (AI4IoT 2023)*, pages 434-440
ISBN: 978-989-758-661-3

Performance of above variants is compared against following state-of-the-art classification algorithms:
Logistic regression
KNN
Decision trees
Gradient boosting
The presentation of the paper is as follows. Second section begins with an analysis of prior research on classification algorithms. Following this, there is discussion about the proposed classification methodology. Next, the time complexities of various models are detailed. Experimental findings are presented in the next section. Finally, the findings of the study and recommendations for future research work are presented.

2 RELATED WORK

An empirical analysis about the effectiveness of supervised learning on high-dimensional data is carried out by (Rich Caruana et al., 2008). The authors implement several machine learning algorithms like support vector machine (SVM), artificial neural network (ANN) and others. These models are evaluated on the basis of performance metrics, namely, accuracy, root mean square error, and area under the ROC metric curve. During the study, 11 binary datasets of very high dimensionality are evaluated and it is concluded that random forest (RF), ANN, SVM and boosted trees outperform all other models. It is also observed that the least performing methods are naive Bayes and perceptron. The study also indicates that boosted trees perform well in lower dimensionality datasets but when it is applied above 400 dimensions, it tends to over fit. (Chongsheng Zhang et al., 2017) carry out an empirical study on various emerging classifiers like extreme learning machine (ELM), sparse representation classifier (SRC) and others. These classifiers are compared with traditional classifiers like random forest, k- nearest neighbors (KNN) etc. During the study, 71 datasets are experimented to validate the effectiveness of the models. The results indicate that the stochastic gradient boosting decision trees perform well in supervised learning. (Jingjun Bi et al, 2018) propose a new machine learning method based on multi class imbalance, namely, Diversified Error Correcting Output Codes (DECOC). To validate the effectiveness of their model, they perform experiments on 17 multi class imbalance datasets. The results indicate that the DECOC achieve best results in terms of accuracy (ACC), area under the ROC Curve (AUC), geometric mean (G-mean) and F- measure. (Amanpreet Singh et al., 2016) compare various supervised machine learning algorithms on the various datasets, on the basis of accuracy, speed, comprehensibility and speed of learning. The authors employ Bayesian networks, naive Bayes, KNN, etc. The authors suggest that choice of an appropriate algorithm depends on the dataset and type of classification problem. The authors conclude from the experimental results that the tree-based algorithms perform better than the rest of the algorithms. According to (Rich Caruana et al., 2006), multiple performance criteria are used to compare learning models in various domains. A model may perform well on one measure but poorly on another. Multiple performance measures assess various trade-offs in prediction. As a result, they evaluate algorithms based on a relatively wide range of performance indicators. The authors compare the ten supervised algorithms using eight distinct performance metrics. They examine the performance indicators before and after using Platt scaling and isotonic regression to calibrate the outputs. They come to the conclusion that calibrated boosted trees outperform other methods in all eight measures. Random Forest is at the second place. Logistic regression and naive Bayes fare the worst. They also find that calibration with either Platt scaling or isotonic regression enhances SVM, stumps, and Naive Bayes performance. (Henry Brighton et al., 2002) start their study by detailing some practical challenges in classification algorithms. The main argument they make is that reduction methods have, historically, been seen as generic solutions to the issue of instance selection. Their studies of, how various schemes function and how well they perform in different contexts, lead them to believe that the success of a scheme is strongly reliant on the structure of the instance-space. They contend that one selection criteria is insufficient to ensure excellent overall performance. They conclude that for the vast majority of classification issues, border instances are crucial to class discrimination. Their algorithm competes with the best effective current methods in 30 fields. (Saksham Trivedi et al., 2021) use ML algorithms in many fields of study. They come to the conclusion that assignment structure has the greatest impact on algorithm selection in machine learning. They assert that SVM and neural networks are more valuable due to their multidimensionality despite the fact that logic systems are ordinarily capable of handling differential/categorical characteristics. For neural network models and SVMs to achieve maximum accuracy, a large sample size is required, whereas NB only requires a small amount of data. Makdah et al., 2019 have noted that under the nominal conditions, the models perform well. Based on the numerical results obtained from the experiments done by the authors it is suggested that the accuracy- sensitivity trade-off is purely determined by the statistical characteristics of the data and cannot be enhanced by adjusting or enhancing the level of complexity of the algorithms. In the paper, the authors have presented a study about trade-off among a binary classification algorithm's accuracy and its susceptibility to uncontrolled modification of data and conclude that a classification algorithm's accuracy can only be maximized at the price of its sensitivity, given a set of moderate technical assumptions. As a result, there is a basic trade-off between the performances of a classification system in conventional and adversarial settings respectively. (Vaishali Gangwar, 2012) present a summary of the categorization of unbalanced data sets. In the study, it is observed that sampling is the most often used strategy to deal with unbalanced data and in case of locally trained classifiers, oversampling outperform under sampling approach but this scenario is inverted in the case of global learning. However, the researcher demonstrate that hybrid sampling strategies outperform oversampling and under sampling. The research suggests that in order to handle uneven data, solutions based on modified support vector machines, rough set-based minority class focused rule learning approaches, and cost sensitive classifiers can be used as an alternative to the classical approach. (José A. Sáez et al., 2013) conduct a comparative analysis of the noise robustness of single classifiers and Multiple Classifier Systems (MCS). The authors attempt to determine the efficacy and robustness of singular classifiers when trained on noise datasets. It is concluded that the robustness of the model against noise depends on the noise level, and in the majority of cases the MCS outperform the individual classifiers. In situations, where the MCS is constructed from heterogeneous classifiers, single classifiers are deemed preferable.

3 PROPOSED METHODOLOGY

The proposed system (Figure 1) constitutes of the following modules: data gathering, label encoding, data type conversion, null value imputation, model building and performance analysis.

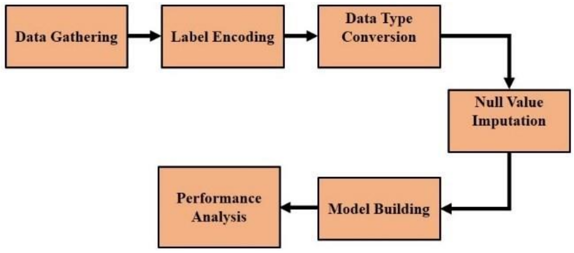

Figure 1: Proposed methodology for the instance-based classifier.

3.1 Data Gathering

In the study, 10 datasets are selected and these datasets are divided into 3 main groups based on the types of features present in the dataset. First group contains datasets having only categorical features, second group datasets contains only numerical features and the last group contains datasets having both categorical and numerical features. Dataset selected in the first group are mushroom, car evaluation and nursery dataset. Mushroom dataset has been collected from the UCI repository and it includes data on mushrooms which have been labelled as either edible or poisonous. It consists of 23 features, including class label and have 8124 instances. Car Evaluation dataset has also been extracted from UCI repository and is a multiclass dataset. It includes data on car acceptability which has been labelled as unacceptable, acceptable, good, and very good condition. It has 6 features including class label and is comprised of 1728 instances. Nursery dataset has also been collected from UCI repository and is a multiclass data set. It includes data to help classify nursery school admission applications, and the target has been recommended, priority, and special priority admission. In total, it consists of 8 features, including class label and has 12960 instances. Dataset selected in the second group are red wine, glass identification, and Pima Indians diabetes dataset. Red wine dataset has been collected from UCI repository and it has data which helps to classify as good or not good. It comprises of 12 features including target feature and the continuous value of target variable of the dataset has been converted to discrete by assuming score of wine greater than 7 as good and rest as not good. Glass identification is a multiclass dataset collected from UCI repository and has data to help classify the type of glass based on features. It has 9 features and constitutes of 214 instances. Pima Indians Diabetes dataset is a binary class dataset that has been gathered using Kaggle. It includes data which helps to identify whether a person gets diabetes within five years of the first physical examination. It constitutes of 8 features and 768 instances. Dataset in the third group includes gender classification, car insurance, from Kaggle and is a binary class dataset. It includes data that helps to classify the gender of an individual. The dataset consists of 8 features out of which 5 are categorical features, 2 are numerical and 1 is the target class and it has a total of 5001 instances. It has 19 features out of which 7 are numerical features and 12 are categorical features, including target variable, and constitutes of 10000 instances. Lung cancer prediction dataset has been collected from Kaggle and it includes data which helps to classify the levels of lung cancer. It is a multiclass dataset with target variable being classified as high, medium and other. It has 26 features, out of which 3 are numerical features and 23 being categorical features, including target variable. It comprises of 1000 instances. HR Analytics dataset has been gathered from Kaggle which includes data to help predict whether the employee is looking for a job or not and the target has been classified into binary values. It has 14 features out of which 3 variables are numerical and other 11 are categorical, including the target variable. It comprises of 19158 instances. Label Encoding is a technique for transforming categorical data into numerical data. In this paper, we have used label encoding to convert the categorical variables into numerical values. We have used it because we have applied machine learning models that require numerical inputs like Logistic Regression, KNN and SVC. Label encoding also helps to conserve the memory requirement for the processing of dataset, as generally most of the categorical variables are defined in a string or object format, which takes more space and processing time as compared to numerical ones. By reducing the memory required to contain categorical variables, label encoding enables machine learning models to analyze larger datasets and more complex models with limited computational resources.

3.2 Data Type Conversion

Data type conversion, also known as type casting, is the process of transforming the data type of a variable or value to a different type. Type conversion of the feature is necessary before applying the model on the dataset, as the proposed model performs different operations on categorical and numerical features, due to which it is necessary to convert categorical features into dtype “category”, before applying the model. All the other Machine Learning models do not require an explicit dtype conversion to perform classification on the data.

3.3 Null Value Imputation

Null value imputation is used to fill in missing or null values in a dataset. It includes replacing the missing values with estimated or imputed values and can be done using a variety of methods. Numerical features are usually imputed by finding mean of the non-missing values of the column and categorical features are imputed by finding the mode of the non-missing values of the column. Imputed null values are required, because they can increase model accuracy by filling in missing values and preventing the loss of crucial data. Algorithms like decision trees are more robust to the presence of null values as compared to Logistic Regression. Presence of null values can affect the KNN algorithm, as it calculates the distance between data points, which can be interfered with by null values. If null values are not handled correctly, they may result in bias or errors. Generally Median imputation is used, when there are significant outliers in the datasets. Since there are no significant outliers present in the dataset, mean imputation has been employed for the datasets containing missing values.

3.4 Model Building

Algorithm 1: Proposed instance-based classifier.

```
SimpleLearning (trainingDataSet, test): class
    For each attributeValue in test
        For each class
    If present in lookup
    Get classCount
        Else
      For each instance in trainingDataSet
      If attributeValue missing
          Ignore
        Else
          Get classCount Increment
          classCount
              Endfor
      Update lookup

  Endfor
    Endfor
   class = max (attributeValue) // leading in more attributes
   If tie
  class = max (totalCount) // leads in total occurrences
          across all attributes
   If tie
       class = any (class)
```

Model is built as per the Algorithm1 given below. Model 1, 2 and 3 are based on the Algorithm1 mentioned below. Model 1 is implemented only for categorical attributes, while model 2 deals with mixed datasets. Model 3 also deals with mixed datasets but incorporates dictionary and lookup features to improve the execution time, which is not there in models 1 and 2.

4 TIME COMPLEXITY ANALYSIS

It is obvious from the given algorithm that the time complexity is O (d*c*n), where d is number of attributes, c is the number of classes and n is the number of instances (training). Table 1 lists time complexity of various models used in our study where, k: number of neighbours, T: number of trees.

Table 1: Model time Complexity.

Algorithm	Time Complexity
K-Nearest Neighbour	O(k*n*d)
Logistic Regression	O(n*d)
Decision Tree	O(n*log(n)*d)
Gradient Boosting	O(T*n*(log(d))
Proposed model	O (d*c*n)

Table 2: Performance on categorical datasets by various models is shown.

DatasetName	Models	Accuracy(in %)	Precision	Recall	F1-Score	Execution Time(in secs)
Mushroom dataset (Binary class)	Logistic Regression	95.63	0.97	0.94	0.955	0.367
	KNN	99.8	0.99	1	0.99	0.310
	Decision Trees	98.15	0.97	0.99	0.98	0.021
	Gradient Boosting	100	1	1	1	0.483
	Model 1	89.66	0.99	0.79	0.88	24.15
	Model 2	89.66	0.99	0.79	0.88	36.096
	Model 3	89.53	0.99	0.79	0.88	2.926
Car Evaluation(Multi class)	Logistic Regression	68.79	0.69	0.69	0.69	0.056
	KNN	93.64	0.94	0.94	0.94	0.053
	Decision Trees	89.6	0.9	0.9	0.9	0.016
	Gradient Boosting	96.24	0.96	0.96	0.96	0.624
	Model 1	73.41	0.73	0.73	0.73	1.186
	Model 2	73.41	0.73	0.73	0.73	2.107
	Model 3	73.41	0.73	0.73	0.73	0.594
Nursery dataset(Multi class)	Logistic Regression	77.16	0.77	0.77	0.77	0.623
	KNN	93.6	0.94	0.94	0.94	0.172
	Decision Trees	87.35	0.87	0.87	0.87	0.018
	Gradient Boosting	98.69	0.98	0.98	0.98	3.217
	Model 1	55.43	0.55	0.55	0.55	13.360
	Model 2	55.43	0.55	0.55	0.55	23.517
	Model 3	42.36	0.42	0.42	0.42	1.247

Table 3: Performance on numerical datasets by various models is shown.

DatasetName	Models	Accuracy(in %)	Precision	Recall	F1-Score	Execution Time (in secs)
Red Wine dataset (Binary class)	Logistic Regression	89.38	0.67	0.26	0.37	16.176
	KNN	89	0.61	0.28	0.39	0.019
	Decision Trees	88.44	0.54	0.31	0.39	0.010
	Gradient Boosting	90.94	0.68	0.49	0.57	0.338
	Model 2	87.5	0.88	1	0.93	5.665
	Model 3	87.5	0.88	1	0.93	6.774
Glass dataset (Multi class)	Logistic Regression	55.81	0.56	0.56	0.56	0.350
	KNN	69.77	0.7	0.7	0.7	0.010
	Decision Trees	67.44	0.67	0.67	0.67	0.009
	Gradient Boosting	72.09	0.72	0.72	0.72	0.733
	Model 2	46.51	0.46	0.46	0.46	0.595
	Model 3	46.51	0.46	0.46	0.46	0.625
Diabetes dataset (Binary class)	Logistic Regression	74.68	0.77	0.46	0.58	0.035
	KNN	67.53	0.61	0.38	0.47	0.013
	Decision Trees	71.43	0.67	0.47	0.55	0.011
	Gradient Boosting	72.73	0.7	0.48	0.57	0.221
	Model 2	68.18	0.65	0.33	0.44	1.883
	Model 3	68.18	0.65	0.33	0.44	2.015

Table 4: Performance on mixed datasets by various models is shown.

DatasetName	Models	Accuracy(in %)	Precision	Recall	F1-Score	Execution Time (in secs)
Gender dataset (Binary class)	Logistic Regression	96.6	0.95	0.97	0.96	0.056
	KNN	97.1	0.97	0.97	0.97	0.047
	Decision Trees	96.4	0.97	0.95	0.96	0.015
	Gradient Boosting	97.2	0.97	0.97	0.97	0.303
	Model 2	93.7	0.89	0.986	0.94	8.1
	Model 3	93.7	0.89	0.986	0.94	7.563
Car Insurance Dataset (Binary class)	Logistic Regression	82.5	0.76	0.65	0.7	0.085
	KNN	79.65	0.69	0.65	0.67	0.185
	Decision Trees	83.75	0.74	0.75	0.746	0.025
	Gradient Boosting	85.8	0.8	0.73	0.77	1.071
	Model 2	68.2	0.68	1	0.81	42.584
	Model 3	68.2	0.68	1	0.81	25.05
Lung Cancer Prediction (Multi class)	Logistic Regression	100	1	1	1	1.308
	KNN	100	1	1	1	0.085
	Decision Trees	100	1	1	1	0.030
	Gradient Boosting	100	1	1	1	1.027
	Model 2	91.5	0.915	0.915	0.915	3.121
	Model 3	91.5	0.915	0.915	0.915	0.677
HR Analytics (Binary class)	Logistic Regression	76.17	0.58	0.25	0.35	0.636
	KNN	70.82	0.35	0.17	0.23	0.191
	Decision Trees	78.24	0.6	0.43	0.5	0.034
	Gradient Boosting	78.63	0.61	0.44	0.51	1.474
	Model 2	74.47	0.74	1	0.85	76.89
	Model 3	74.47	0.74	1	0.85	46.448

5 EXPERIMENTAL FINDINGS

In this study we have incorporated four machine learning models which are KNN, Decision Tree, Logistic Regression, and Gradient Boost. KNN is used in the study as it is based on lazy learning approach and is simple and doesn't make any assumptions about the distribution of the data. However, it can be computationally expensive for large datasets and is also sensitive to the choice of k. Logistic regression was used as it is easy to implement and understand relationship between features and target variables. Decision Tree was implemented in the study as it can handle non-linear relationship in data and doesn't make any assumption about the distribution of data. However, it is prone to overfitting and doesn't work well on small datasets. Gradient Boosting model was used in the study as it can handle non-linear relationships and is less prone to outliers. However, it needs tuning of hyper parameters and can also overfit if the model is too complex. As we can see from the tables 2 and 4 the execution time for the model-3 has always been less compared to the model 1 and 2. Model 3 is having a reduced time largely because of the reason that it creates a dictionary and it updates the test data of majority vote of each attribute in the dictionary, due to which after some test points the dictionary will mostly have all the majority vote values for each attribute and it is not needed to calculate the majority vote of the attributes again, hence the testing time gradually decreases . From the above tables existing machine learning models like Logistic Regression, KNN, Decision Tree and Gradient Boosting and sometimes we can see the models outperforming the KNN and logistic regression models, but we can also observe the dip in the precision values because of the class imbalance. When there is class imbalance in the dataset then the algorithm votes for is high in number because of which the minority class predictions are outnumbered by the majority classes. Due to which there is a decrease in correct classification of the minority classes. All the algorithms are executed in the Google Collab environment which has 12GB RAM.

6 CONCLUSIONS

The results obtained from the implementation of the proposed three model classifiers have been comparable to those from the base line classifiers. In some cases, the proposed model has performed better while in other cases there has been some dip in performance metrics due to the fact of imbalance and high bias of output classes of datasets. From the performance metrics, it has been observed that for some datasets, where there is high bias towards one class of output, the model under performs in terms of performance metrics precision, recall and F1 score. In the third model, there has been an improvement in terms of computational time taken for training the model. Another important observation is significantly higher recall of the proposed classifier in certain cases, which may need further investigation. Even precision of proposed model is also noteworthy in certain cases as compared to other models. One of the future prospects of this paper could be to perform various data sampling techniques to prevent the model from over-fitting. Although the training time of the model three has been considerably dropped, there is still a scope for improvement in computational time complexity of the model, by leveraging various parallel architectures. In future, the custom models can be tested on higher dimension datasets as well.

REFERENCES

“UCIMachine Learning Repository.” n.d. Archive.ics.uci.edu.https://archive.ics.uci.edu/dataset/73/mushroom.

“UCI Machine Learning Repository.” n.d. Archive.ics.uci.edu. Accessed March 17, 2023. https://archive.ics.uci.edu/dataset/19/car+evaluatio

“UCI Machine Learning Repository.” n.d. Archive.ics.uci.edu. Accessed March 17, 2023. https://archive.ics.uci.edu/dataset/76/nursery.

“UCIMachine Learning Repository.” n.d. Archive.ics.uci.edu.https://archive.ics.uci.edu/dataset/186/wine+quality.

“UCI Machine Learning Repository.” n.d. Archive.ics.uci.edu. Accessed June 17, 2023. https://archive.ics.uci.edu/dataset/42/glass+identification.

“Pima Indians Diabetes Database.” n.d. Www.kaggle.com.https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

“Gender Classification Dataset.” n.d. Www.kaggle.com.https://www.kaggle.com/datasets/elakiricoder/gender-classification-dataset.

“Car Insurance Data.” n.d. Www.kaggle.com. https://www.kaggle.com/datasets/sagnik1511/car-insurance-data.

“Lung Cancer Prediction.” n.d. Www.kaggle.com. https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link.

“HR Analytics: Job Change of Data Scientists.” n.d. Www.kaggle.com.https://www.kaggle.com/datasets/arashnic/hr-analytics-job-change-of-data-scientists?select=aug_ train.csv.

Bi, Jingjun, and Chongsheng Zhang (2018). "An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme" Knowledge-Based Systems 158: 81-93.

Caruana, Rich, Nikos Karampatziakis, and Ainur Yessenalina (2008). "An empirical evaluation of supervised learning in high dimensions" In Proceedings of the 25th international conference on Machine learning, pp. 96-103.

Zhang, Chongsheng, Changchang Liu, Xiangliang Zhang, and George Almpanidis (2017). "An up-to-date comparison of state-of-the-art classification algorithms" Expert Systems with Applications 82: 128-150.

A. Singh, N. Thakur and A. Sharma (2016) "A review of supervised machine learning algorithms" 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, pp. 1310-1315.

Caruana, Rich, and Alexandru Niculescu-Mizil (2006). "An empirical comparison of supervised learning algorithms" In Proceedings of the 23rd international conference on Machine learning, pp. 161-168.

Brighton, Henry, and Chris Mellish (2002). "Advances in instance selection for instance-based learning algorithms" Data mining and knowledge discovery 6: 153-172.

Trivedi, Saksham, Balwinder Kaur Dhaliwal, and Gurpreet Singh (2021). "A review paper on a comparative study of supervised learning approaches" In 2021 International Conference on Computing Sciences (ICCS), pp. 95-100. IEEE.

Al Makdah, Abed Al Rahman, Vaibhav Katewa, and Fabio Pasqualetti (2019)."A fundamental performance limitation for adversarial classification" IEEE Control Systems Letters 4, no. 1: 169-174.

Gangwar, Vaishali (2012)."An overview of classification algorithms for imbalanced datasets" International Journal of Emerging Technology and Advanced Engineering 2, no. 4: 42-47.

Sáez, José A., Mikel Galar, Julián Luengo, and Francisco Herrera (2013)."Tackling the problem of classification with noisy data using multiple classifier systems: Analysis of the performance and robustness" Information Sciences 247: 1-20.

Sarker, I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160 (2021). https://doi.org/10.1007/s42979-021-00592-x

Tammy Jiang, Jaimie L. Gradus, Anthony J. Rosellini, Supervised Machine Learning: A Brief Primer, Behavior Therapy, Volume 51, Issue 5, 2020, Pages 675 687, ISSN 0005-7894, https://doi.org/10.1016/j.beth.2020.05.002. (https://www.sciencedirect.com/science/article/pii/S0005789420300678