











Article

Redundancy Is Not Necessarily Detrimental in Classification Problems

Sebastián Alberto Grillo ¹, José Luis Vázquez Noguera ^{1,*}, Julio César Mello Román ^{1,2,3}, Miguel García-Torres ^{1,4}, Jacques Facon ⁵, Diego P. Pinto-Roa ^{1,2,3}, Luis Salgueiro Romero ⁶, Francisco Gómez-Vela ⁴, Laura Raquel Bareiro Paniagua ¹ and Deysi Natalia Leguizamón Correa ¹

- ¹ Computer Engineer Department, Universidad Americana, Asunción 1206, Paraguay; sebastian.grillo@ua.edu.py (S.A.G.); juliomello@pol.una.py (J.C.M.R.); mgarcia@upo.es (M.G.-T.); dpinto@pol.una.py (D.P.P.-R.); laura.bareiro@ua.edu.py (L.R.B.P.); deysi.leguizamón@ua.edu.py (D.N.L.C.)
- ² Facultad Politécnica, Universidad Nacional de Asunción, San Lorenzo 111421, Paraguay
- ³ Facultad de Ciencias Exactas y Tecnológicas, Universidad Nacional de Concepción, Concepción 010123, Paraguay
- ⁴ Data Science and Big Data Lab, Universidad Pablo de Olavide, 41013 Seville, Spain; fgomez@upo.es
- ⁵ Department of Computer and Electronics, Universidade Federal do Espírito Santo, São Mateus 29932-540, Brazil; jacques.facon@ufes.br
- ⁶ Signal Theory and Communications Department, Universitat Politècnica de Catalunya, 8034 Barcelona, Spain; luis.fernando.salgueiro@upc.edu
- * Correspondence: jose.vazquez@ua.edu.py



Citation: Grillo, S.A.; Noguera, J.L.V.; Mello Román, J.C.; García-Torres, M.; Facon, J.; Pinto-Roa, D.P.; Salgueiro Romero, L.; Gómez-Vela, F.; Paniagua, L.R.B.; Correa, D.N.L. Redundancy Is Not Necessarily Detrimental in Classification Problems. *Mathematics* **2021**, *9*, 2899. <https://doi.org/10.3390/math9222899>

Academic Editor: Liangxiao Jiang

Received: 24 September 2021

Accepted: 5 November 2021

Published: 15 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: In feature selection, redundancy is one of the major concerns since the removal of redundancy in data is connected with dimensionality reduction. Despite the evidence of such a connection, few works present theoretical studies regarding redundancy. In this work, we analyze the effect of redundant features on the performance of classification models. We can summarize the contribution of this work as follows: (i) develop a theoretical framework to analyze feature construction and selection, (ii) show that certain properly defined features are redundant but make the data linearly separable, and (iii) propose a formal criterion to validate feature construction methods. The results of experiments suggest that a large number of redundant features can reduce the classification error. The results imply that it is not enough to analyze features solely using criteria that measure the amount of information provided by such features.

Keywords: feature selection; feature construction; classification

1. Introduction

In the classification, the quality of information in the features is essential to building a high-quality predictive model. Furthermore, the rapid advances in data acquisition and storage technologies have created high-dimensional data. However, noise, non-informative features, and redundancy, among other issues, make the classification task challenging [1]. Therefore, selecting suitable features is an important task, as a preliminary step, for building highly predictive classifiers [2].

To reduce dimensionality, there are two main approaches—feature selection and feature construction. Feature selection selects a subset of features from the input to reduce the effects of noise or irrelevant features, while still providing good prediction results [2]. In contrast, feature construction refers to the task of transforming a given set of input features to generate a new set of more predictive features [3].

According to [2], feature selection can be divided into three major categories depending on the evaluation criteria—filter, wrapper, and embedded. Filter methods use intrinsic properties of the data to select a subset of features and are applied as a preprocessing task [4]. Wrappers, in contrast, use learning to guide the search. The learning bias is included in the search and, therefore, they achieve better results. However, they are com-

putationally expensive [5,6] and cause overfitting [7]. Finally, embedded methods perform the search at the same time the model is learned.

However, we can also classify the feature selection methods according to the strategy to search for subsets of features, which are divided into exponential search, sequential search, and random search [8]. The exponential search consists of the exhaustive evaluation of all possible subsets, which makes it impractical most of the time. The sequential search consists of the application of a local search method with a hill descent strategy [9,10]. The use of such strategies means that the search is stuck in a local optima. Finally, we have random search strategies that consist of the application of metaheuristic optimization algorithms [11–18].

Despite the success of feature selection techniques, a good feature space is a prerequisite for achieving high performance in the classification. In this sense, feature construction aims to engineer new features to detect the hidden relations of the original features [19,20]. New features are constructed based on the relations of the original ones pursuing a more meaningful feature space capable of achieving a more accurate classifier [3]. As in the case of feature selection, in feature construction, we can find three approaches: filter, wrapper, and embedded methods [21]. Among the main approaches for constructing features, we have (i) methods based on decision trees [22,23], (ii) evolutionary meta-heuristics [24,25], (iii) the application of inductive logic programming [26,27], (iv) methods that use annotations with the training set [28], and (v) unsupervised methods such as clustering [29], PCA [30], or SVD [31].

In this work, we study the relationship of feature construction and assumptions applied in selecting those features. Denote—as redundant—a subset of features that do not provide more information than what exists in the other features. We are particularly interested in analyzing the assumption that minimizing the number of redundant features is best for classification problems. Especially how the defined features can affect the capacity of a model required to perform the classification. We first present a mathematical framework for modeling feature construction and selection for classification problems with discrete features. Second, we show that there are datasets where small feature subsets can be much more complex than large feature subsets. We denote complexity concerning the capacity that the model requires to classify the problem and highlight the linearly separable problems as the least complex. This construction violates the assumption that fewer features with equal or more information are better than many features. Third, we extend the analysis of feature construction using monomials of degree k [32] and conclude that this method tends to produce linearly separable binary classification problems as k grows. Therefore, we propose that one way to validate feature construction methods is by analyzing whether the classification problems tend to become linearly separable with the iterative application of the method. Finally, we apply the construction of features with monomials of degree k in real and artificial datasets, where we apply the following classification algorithms, naive Bayes [33], logistic regression [34], KNN [35], PART [36], JRIP [37], J48 [38] and random forest [39]. Experiments show that even though redundant features grow extensively, the score increases or does not decrease too much. Therefore, both theoretical and experimental evidence agree that the criterion of choosing minimum feature subsets is not always correct. This is because the assumption considers only the information about the features but not the complexity of the classification problem.

The contributions of this work can be synthesized in the following items: (a) showing that the redundancy of features can reduce data complexity, (b) developing a theoretical framework to model construction and selection of features and, (c) proposing a mathematical criterion to validate feature construction methods. The experiments performed suggest that the presence of redundant features does not necessarily prejudice classification tasks.

This work is organized into the following sections. Section 2 presents the mathematical formulation used to describe the theoretical results. Section 3 introduces basic ideas with simple examples, while Section 4 formalizes those ideas to more general results. Section 5

shows the experimental results, and finally, Section 6 presents a discussion of all results obtained.

2. A Mathematical Model for Feature Selection and Construction

In this section, we present a formal framework for the mathematical analysis of feature selection and construction. Let $\{A_i\}$ be a finite sequence of finite sets in \mathbb{R} and another finite set C , where each A_i is denoted as feature i and C is the set of possible classes. Taking $\mathcal{A} = A_1 \times A_2 \times \dots \times A_n$, we consider a probability distribution \mathcal{P} over $\mathcal{A} \times C$, we denote $\mathbb{P}[\cdot]$ and $\mathbb{P}[\cdot|\cdot]$ as the probability and conditional probability determined by \mathcal{P} , respectively. Notice that we may generate a dataset using distribution \mathcal{P} , where each record is an element from $\mathcal{A} \times C$ and we denote \mathcal{P} as a dataset distribution. Denote the sequence $\{\hat{A}_i\}$, such that $\hat{A}_i = A_i$ for $i \leq n$ and $\hat{A}_{n+1} = C$. Let $\{S_i\}$ be a subsequence of $\{\hat{A}_i\}$, we denote (i) $\mathcal{S} = S_1 \times S_2 \times \dots \times S_m$, (ii) if $s \in \mathcal{S}$ then s is denoted as a pattern of \mathcal{S} and (iii) $E_{\mathcal{S}}^{\mathcal{P}}(x)$ is denoted as the event where we sample an instance such that $s = x$ for a pattern s of \mathcal{S} according to distribution \mathcal{P} . We say that s is a not-null pattern of \mathcal{S} if $\mathbb{P}[E_{\mathcal{S}}^{\mathcal{P}}(s)] > 0$.

Notice that our definition of the dataset distribution is general enough for a dataset or its real distribution. For example, given the dataset distribution \mathcal{P} in Table 1, we can take $A_1 = \{1, 2, 3\}$, $A_2 = \{1, 2\}$, $A_3 = \{0, 1, 2, 3\}$, and $C = \{0, 1\}$. As $\mathcal{S} = S_1 \times S_3$ represents all possible values taken by the first and third feature, if $s = (1, 1)$ is a pattern of \mathcal{S} , then $E_{\mathcal{S}}^{\mathcal{P}}(x) = \{(1, 1, 1, 0), (1, 2, 1, 0), (1, 1, 1, 1), (1, 2, 1, 1)\}$ is the event where the first and third features have value one. Notice that s is a not-null pattern because $\mathbb{P}[E_{\mathcal{S}}^{\mathcal{P}}(s)] = 2/5$.

Table 1. Simple example of dataset distribution.

Att. 1	Att. 2	Att. 3	Class
1	1	0	0
1	2	1	0
1	2	1	1
2	1	2	1
3	1	3	0

The following definition formalizes the notion of patterns that do not contradict each other.

Definition 1. Let $B = \{B_i\}$ and $D = \{D_i\}$ be sub-sequences of $\{A_i\}$, we denote $\mathcal{B} = B_1 \times B_2 \times \dots \times B_p$ and $\mathcal{D} = D_1 \times D_2 \times \dots \times D_q$. Taking $b = (b_1, b_2, \dots, b_p) \in \mathcal{B}$ and $d = (d_1, d_2, \dots, d_q) \in \mathcal{D}$, we say that b and d are congruent patterns, if b and d are not distinct in the features of $\{A_i\}$ preserved by both $B = \{B_i\}$ and $D = \{D_i\}$.

For example, take the dataset distribution \mathcal{P} of Table 1, $B = \{A_1, A_2\}$ and $D = \{A_2, A_3\}$. We have that $b = (1, 2) \in \mathcal{B}$ and $d = (2, 1) \in \mathcal{D}$ are congruent patterns, because they have the same value in their single shared feature. However, if $\hat{d} = (1, 2) \in \mathcal{D}$, then b and \hat{d} are not congruent patterns, because both have different values given the second feature of the dataset.

As a dataset distribution \mathcal{P} may not be consistent (inconsistent), we define a function $f_{\mathcal{P}} : \mathcal{A} \rightarrow C$, where $\mathbb{P}[E_C^{\mathcal{P}}(c) | E_{\mathcal{A}}^{\mathcal{P}}(a)] = \max_i \{\mathbb{P}[E_C^{\mathcal{P}}(i) | E_{\mathcal{A}}^{\mathcal{P}}(a)]\}$ for all not-null patterns $a \in \mathcal{A}$. Notice that an inconsistent dataset distribution always has classification error because a classifier does not have enough features, then $f_{\mathcal{P}}$ gives the category that minimize error for any configuration of features. If we consider the dataset distribution of Table 1, we must define a $f_{\mathcal{P}}$, such that $f_{\mathcal{P}}(1, 1, 0) = 0$, $f_{\mathcal{P}}(2, 1, 2) = 1$ and $f_{\mathcal{P}}(3, 1, 3) = 0$; however for any other pattern $a \in \mathcal{A}$ we can take 0 or 1 for $f_{\mathcal{P}}$.

Definition 2. Let \mathcal{P} be a dataset distribution, $B = \{B_i\}$ a sub-sequence of sequence $A = \{A_i\}$ and $\mathcal{B} = B_1 \times B_2 \times \dots \times B_p$. The subsequence B of features is complete for \mathcal{P} if satisfies that for all class c and all congruent not-null patterns a, b of \mathcal{A}, \mathcal{B} , respectively, we have:

$$\mathbb{P}\left[E_C^{\mathcal{P}}(c) \mid E_{\mathcal{A}}^{\mathcal{P}}(a)\right] = \mathbb{P}\left[E_C^{\mathcal{P}}(c) \mid E_{\mathcal{B}}^{\mathcal{P}}(b)\right].$$

Definition 2 formalizes the notion of a subset of features with the same amount of information as all features as a whole. This notion of information considers that the subset of features is sufficient to estimate the class with the same probability as the original set of features.

Definition 3. Maintaining the same terms of Definition 2. Let $\hat{B}^k = \{\hat{B}_i\}$ be a sub-sequence of sequence B without the term A_k and $\hat{\mathcal{B}}^k = \hat{B}_1 \times \hat{B}_2 \times \dots \times \hat{B}_q$. The subsequence B of features is non-redundant for \mathcal{P} , if it satisfies that for all k there is some class c , and some not-null congruent patterns b, \hat{b} of $\mathcal{B}, \hat{\mathcal{B}}^k$, respectively, such that:

$$\mathbb{P}\left[E_C^{\mathcal{P}}(c) \mid E_{\mathcal{B}}^{\mathcal{P}}(b)\right] \neq \mathbb{P}\left[E_C^{\mathcal{P}}(c) \mid E_{\hat{\mathcal{B}}^k}^{\mathcal{P}}(\hat{b})\right].$$

Definition 3 formalizes the notion of a subset of features where each feature provides information that does not exist in other features of the subset. This notion of information considers that if we eliminate a feature from the subset, we will not obtain the same probability of obtaining a class. Under this definition of a non-redundant subset of features, we can say that the other features of the dataset are redundant because they can be eliminated without losing information in the dataset. We formulate Definition 4 for redundant features.

Definition 4. Maintaining the same terms of Definition 2. Let $\hat{A} = \{\hat{A}_i\}$ be a subsequence of A , obtained by eliminating the features of a subsequence B from A . The subsequence B of features is redundant for \mathcal{P} if it satisfies that for all class c and not-null congruent patterns a, \hat{a} of $\mathcal{A}, \hat{\mathcal{A}}$ respectively, we have:

$$\mathbb{P}\left[E_C^{\mathcal{P}}(c) \mid E_{\mathcal{A}}^{\mathcal{P}}(a)\right] = \mathbb{P}\left[E_C^{\mathcal{P}}(c) \mid E_{\hat{\mathcal{A}}}^{\mathcal{P}}(\hat{a})\right].$$

Taking the dataset distribution \mathcal{P} of Table 1 again, we can see that $\{A_1, A_2\}$ and $\{A_3\}$ are complete and non-redundant for \mathcal{P} . Sub-sequences composed by individual non-constant features like $\{A_1\}$ and $\{A_2\}$ are non-redundant, but not complete for \mathcal{P} . Finally, sub-sequences like $\{A_1, A_2, A_3\}$, $\{A_2, A_3\}$ and $\{A_1, A_3\}$ are complete, but not non-redundant for \mathcal{P} .

Definition 5. Let \mathcal{P} be a dataset distribution over $\mathcal{A} \times C$. Let B_i be a sequence of finite sets, $\mathcal{B} = B_1 \times B_2 \times \dots \times B_p$ and $\hat{\mathcal{A}} = \mathcal{A} \times \mathcal{B}$. We say that a dataset distribution \mathcal{Q} over $\hat{\mathcal{A}} \times C$ is an extension of \mathcal{P} , if (i) $\mathbb{P}\left[E_{\mathcal{S}}^{\mathcal{Q}}(s)\right] = \mathbb{P}\left[E_{\mathcal{S}}^{\mathcal{P}}(s)\right]$ for all $\mathcal{S} = S_1 \times S_2 \times \dots \times S_m$ and $s \in \mathcal{S}$, where $S \subset \{A_i\} \cup C$ and (ii) for all not null pattern $a \in \mathcal{A}$ there is some pattern $b \in \mathcal{B}$ such that $\mathbb{P}\left[E_{\mathcal{B}}^{\mathcal{Q}}(b) \mid E_{\mathcal{A}}^{\mathcal{Q}}(a)\right] = 1$.

Definition 5 formalizes the notion of feature construction. It consists of a new dataset distribution whose set of features contains the set of features of the original dataset with the same distribution according to the first property. However, according to the second property, the new distribution also contains new features whose values are entirely determined by the shared features.

Following the dataset distribution \mathcal{P} of Table 1, we denote a dataset distribution $\hat{\mathcal{P}}$ from Table 1 where we eliminate the feature A_3 . Notice that \mathcal{P} is an extension of $\hat{\mathcal{P}}$, because (i) as $\hat{\mathcal{P}}$ is \mathcal{P} without a feature, then they have the same probabilities for the common features and (ii) if we know the values of features 1 and 2, then we know the value of feature 3 with probability 1 for any not-null pattern.

3. Features: Selection vs. Construction

In this section, we use the mathematical notions defined above to compare selection with feature construction. In this sense, feature selection is denoted as an elimination of features, while feature construction is denoted as incorporating new features.

Feature selection methods that do not involve the classifier in the selection are called filter methods. These methods are based on applying some measure that seeks to obtain a subset of features, which contains the same amount of information as the original set but without any redundancy. The literature reports several of these methods; however, they believe that a non-redundant set of features should be as small as possible. This condition can be mathematically described as obtaining a complete and non-redundant sub-sequence of features B for \mathcal{P} , where we minimize $|B|$.

Mathematically we can define the construction of features from a dataset distribution \mathcal{P} as any extension of \mathcal{P} . Feature construction consists of computing new features from the original features. If the result ends up with more features than the original, we come across a method contrary to the minimization criterion of the feature selection by filtering methods.

One of the principles of feature selection by the filtering methods is that redundancy in features is detrimental. We refer to a redundant feature in the sense that all the information existing in the feature can be obtained from a subset of features that does not contain the feature itself. In that sense, the construction of features without a subsequent selection of features only produces redundant features. Formally, we are saying that if \mathcal{Q} is an extension of \mathcal{P} as constructed in Definition 5, then $\mathbb{P}[E_C^{\mathcal{P}}(c) | E_{\mathcal{A}}^{\mathcal{P}}(a)] = \mathbb{P}[E_C^{\mathcal{Q}}(c) | E_{\hat{\mathcal{A}}}^{\mathcal{Q}}(\hat{a})]$ for all $c \in C$ and all not-null congruent patterns a, \hat{a} of $\mathcal{A}, \hat{\mathcal{A}}$, respectively. In other words, although pattern a has extra features to \hat{a} , that does not modify the probabilities of obtaining any class c ; therefore, the extra features do not provide information.

The notion of feature construction introduced by Definition 5 does not add more information because the original features define the new features entirely. Therefore, we are interested in knowing what else can be provided by new features in case these features do not have more information than what already exists.

We analyze a simple example of a classification algorithm interacting with a constructed feature before presenting theorems with more general results. First, we consider the distribution of Table 2, where we assume that the original features are 1 and 2. For each pattern $a \in \mathcal{A}$, feature 3 is defined as $a_3 = (a_1)^2$. Second, we consider a classifier based on the logistic model. If we denote $\mathcal{L}(x) = \frac{1}{1+e^{-x}}$ and the internal parameters or weights $v_0, v_1, v_2 \in \mathbb{R}$, the logistic model applied to the original features of pattern $a \in \mathcal{A}$ outputs 1 if $\mathcal{L}(v_0 + a_1v_1 + a_2v_2) > \frac{1}{2}$ and 0 otherwise. Denoting another parameter $v_3 \in \mathbb{R}$, the logistic model applied to all features of pattern $a \in \mathcal{A}$ outputs 1 if $\mathcal{L}(v_0 + a_1v_1 + a_2v_2 + a_3v_3) > \frac{1}{2}$ and 0 otherwise. Notice that in Figure 1, if we apply the logistic model in the original features, we obtain a linear classifier on the plane for features 1–2 that cannot give the correct class to all instances. Therefore this first model has under-fitting problems. However, if we take the second logistic model with the parameters $v_0 = 17/4, v_1 = -4, v_2 = 0$, and $v_3 = 1$, we obtain a non-linear model over the plane for features 1–2 with the region between Att. 1 = 1.5 and Att. 2 = 2.5 for class 0 and the rest of the plane for class 1. This second logistic model is equivalent to a third logistic model applied to the original features of pattern $a \in \mathcal{A}$ that outputs 1 if $\mathcal{L}(v_0 + a_1v_1 + a_2v_2 + (a_1)^2v_3) > \frac{1}{2}$ and 0 otherwise. We say that both logistic models are equivalent because they partition the plane of features 1–2 exactly as Figure 1 shows. In both the second and third models there is an extra parameter v_3 that modulates the non-linearity in the plane of features 1–2. The second model is a linear model over the space produced by the features 1–3 and behaves non-linearly in the plane of features 1–2 due to feature 3. Instead, the third model is an inherently nonlinear model for features 1–2 for v_3 distinct to 0. Therefore, the construction of features can increase the representation capacity of the model and solve under-fitting problems like the one we observed with the first model.

Table 2. Example of linear separability thanks to a new redundant feature.

Att. 1	Att. 2	Att. 3	Class
1	0	1	1
2	0	4	0
2	1	4	0
3	0	9	1

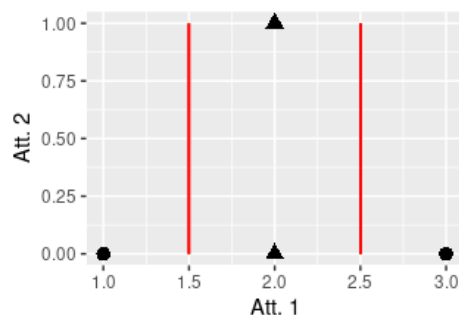


Figure 1. Graph corresponding to Table 2 without feature 3.

4. A Theoretical Analysis of Feature Construction

In this section, we present results that generalize what was stated in Section 3. The following theorem refutes the idea that the fewer features we use without losing information, the better for the classification problem.

Theorem 1. Let \mathcal{P} be a non-constant dataset distribution over $\mathcal{A} \times \{0, 1\}$, whose set of features $\{A_i\}$ is non-redundant in \mathcal{P} . Take p as the total number of non-null patterns in \mathcal{P} whose value by $f_{\mathcal{P}}$ is the minority class between zero and one. For all integer m in $[p, n + 1]$ there is a set of m features $\{B_i\}$ and an extension \mathcal{Q} of \mathcal{P} , such that (i) \mathcal{Q} is a distribution over $\hat{\mathcal{A}} \times \{0, 1\}$ where $\hat{\mathcal{A}} = \mathcal{A} \times \mathcal{B}$, (ii) $\{B_i\}$ is a non-redundant set of features in \mathcal{Q} , (iii) there is a linear classifier that computes $f_{\mathcal{P}}(a)$ from b , if $\hat{b} = (a, b)$ is a not-null pattern of $\hat{\mathcal{A}}$.

Proof. Let \mathcal{N} be the set of not-null patterns of \mathcal{A} according to \mathcal{P} . We denote a partition $\{N_i\}$ of \mathcal{N} of size m , where each N_i contains a pattern with value one and a pattern with value zero according $f_{\mathcal{P}}$. We also take $B_i = \{0, 1\}$ for all i . Then we construct \mathcal{Q} : for each $a \in \mathcal{N}$ we have $(a, b) \in \hat{\mathcal{A}}$, such that: if $a \in N_k$ then $b_k = f_{\mathcal{P}}(a)$ and $b_i = 0$ for all $i \neq k$. As $f_{\mathcal{P}}(a)$ is fully determined by a and b is fully determined by $f_{\mathcal{P}}(a)$, then b is fully determined by a and \mathcal{Q} is an extension of \mathcal{P} .

For the second property, we denote $\hat{\mathcal{B}}^i = B_1 \times \dots \times B_{i-1} \times B_{i+1} \times \dots \times B_m$ and three patterns $b, \tilde{b} \in \mathcal{B}, \hat{b} \in \hat{\mathcal{B}}^i$. We take b, \hat{b} with all terms zero and \tilde{b} with all terms zero except $\tilde{b}_i = 1$. Notice that \hat{b} is congruent to the other patterns and all are non-null patterns, this implies that $E_{\mathcal{B}}^{\mathcal{Q}}(b), E_{\mathcal{B}}^{\mathcal{Q}}(\tilde{b}),$ and $E_{\mathcal{B}}^{\mathcal{Q}}(\hat{b})$ are events with non-zero probability. Then we have:

$$\mathbb{P}\left[E_C^{\mathcal{Q}}(1) \mid E_{\mathcal{B}}^{\mathcal{Q}}(b)\right] = \frac{\mathbb{P}\left[E_C^{\mathcal{Q}}(1) \cap E_{\mathcal{B}}^{\mathcal{Q}}(b)\right]}{\mathbb{P}\left[E_{\mathcal{B}}^{\mathcal{Q}}(b)\right]} \tag{1}$$

and:

$$\mathbb{P}\left[E_C^{\mathcal{Q}}(1) \mid E_{\hat{\mathcal{B}}^i}^{\mathcal{Q}}(\hat{b})\right] = \frac{\mathbb{P}\left[E_C^{\mathcal{Q}}(1) \cap E_{\hat{\mathcal{B}}^i}^{\mathcal{Q}}(\hat{b})\right]}{\mathbb{P}\left[E_{\hat{\mathcal{B}}^i}^{\mathcal{Q}}(\hat{b})\right]} = \frac{\mathbb{P}\left[E_C^{\mathcal{Q}}(1) \cap E_{\mathcal{B}}^{\mathcal{Q}}(b)\right] + \mathbb{P}\left[E_C^{\mathcal{Q}}(1) \cap E_{\mathcal{B}}^{\mathcal{Q}}(\tilde{b})\right]}{\mathbb{P}\left[E_{\mathcal{B}}^{\mathcal{Q}}(b)\right] + \mathbb{P}\left[E_{\mathcal{B}}^{\mathcal{Q}}(\tilde{b})\right]}. \tag{2}$$

As $\frac{x}{y} < \frac{z}{w}$ implies that $\frac{x}{y} < \frac{x+z}{y+w}$ for real positive numbers x, y, z, w and:

$$\frac{\mathbb{P}\left[E_C^Q(1) \cap E_B^Q(b)\right]}{\mathbb{P}\left[E_B^Q(b)\right]} < \frac{\mathbb{P}\left[E_C^Q(1) \cap E_B^Q(\hat{b})\right]}{\mathbb{P}\left[E_B^Q(\hat{b})\right]}. \tag{3}$$

Thus, we have:

$$\mathbb{P}\left[E_C^Q(1) \mid E_B^Q(b)\right] < \mathbb{P}\left[E_C^Q(1) \mid E_B^Q(\hat{b})\right]. \tag{4}$$

For the last property, we need to construct a logistic model that outputs one if $\mathcal{L}(\sum_i b_i) > \frac{1}{2}$ and zero otherwise. Notice that this linear classifier computes $f_{\mathcal{P}}(a)$ for all not-null pattern $\hat{b} = (a, b)$ from \mathcal{Q} . \square

Notice that $\{B_i\}$ can be much bigger than $\{A_i\}$. However, inferring the category labels from $\{A_i\}$ can be as complex as we want, at the same time that selecting the bigger set $\{B_i\}$ instead we will have a problem that is solved by a linear classifier. Therefore, a feature selection method would choose $\{A_i\}$ over $\{B_i\}$ under the criterion of minimizing the number of features.

Although we refer to complexity, there is no single measure of complexity for classification problems [40]. However, it is observed that classifiers that use a single variable, artificial neural networks of a single neuron, and the simplest SVM models are linear classifiers. Additionally, linear classifiers have a VC dimension of the value of only two [41]. Therefore, for our purpose, we consider linearly separable sets as those with less complexity.

Theorem 1 shows an extreme case where feature construction breaks a standard criterion for feature selection methods. However, theorem-proof does not present a practical method for feature construction because we can only build the features in the training set. We note that to construct features $\{B_i\}$, we must know in advance the most probable class for each pattern in \mathcal{A} . That is to say, first solve the classification problem only with the features of $\{A_i\}$, which does not make sense. Therefore, we will now study a standard method for constructing features.

The following definition generalizes the construction of features using monomials, which was used as an example in Section 3. The idea is that there is a feature equivalent to each monomial of degree less than or equal to k from the original features.

Definition 6. Taking same terms from Definition. We denote \mathcal{P}^k as a k -monomial extension of \mathcal{P} and $\mathcal{A}(k)$ as the product of features of \mathcal{P}^k , if (i) for each i , there is a monomial function $f : \mathcal{A}(k) \rightarrow \mathbb{R}$ of grade equal or less than k , such that $\hat{a}_i = f(\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n)$ for each not-null pattern $\hat{a} \in \mathcal{A}(k)$ and (ii) for each monomial function $f : \mathcal{A} \rightarrow \mathbb{R}$ of grade equal or less than k there is some i , such that $\hat{a}_i = f(\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n)$ for each not-null pattern $\hat{a} \in \mathcal{A}(k)$.

For example, suppose that the dataset distribution \mathcal{P} has three features and denote (a_1, a_2, a_3) as a pattern for those features. Then, a pattern from \mathcal{P}^2 could be of the form $(a_1, a_2, a_3, a_1^2, a_2^2, a_3^2, a_1a_2, a_1a_3, a_2a_3)$ and a pattern from \mathcal{P}^3 could be of the form $(a_1, a_2, a_3, a_1^2, a_2^2, a_3^2, a_1a_2, a_1a_3, a_2a_3, a_1^3, a_2^3, a_3^3, a_1^2a_2, a_1^2a_3, a_2^2a_3, a_1a_2^2, a_1a_3^2, a_2a_3^2)$.

Notice that Definition 6 does not give an explicit order for the new features, however Definition 5 just guarantees that the first n features of \mathcal{P}^k are the original features of \mathcal{P} . Then the features i in \mathcal{P}^k for $i > n$ are in function of the first n features in \mathcal{P}^k (that also are the features of \mathcal{P}).

The following theorem describes how the feature construction method described in Definition 6 can reduce the complexity of the classification problem.

Theorem 2. For all dataset distribution \mathcal{P} over $\mathcal{A} \times \{0, 1\}$, there is some k such that some linear classifier computes $f_{\mathcal{P}^k}$ from the not-null patterns of \mathcal{P}^k .

Proof. Let \mathcal{P} be a dataset distribution over $\mathcal{A} \times \mathcal{C}$ whose features A_i have more than one possible value, without loss of generalization. We denote (i) the minimum absolute difference between values in the feature A_i as β_i , (ii) the difference between the maximum and minimum values in the feature A_i as δ_i and (iii) the maximum δ_i/β_i as D . Then, from $a \in \mathcal{A}$ we define the function $g(a) = \sum_i a_i(3D)^{i-1}$, which is a polynomial of grade 1 on the terms of a . Notice that g is an injective function if we take \mathcal{A} as the domain. We denote P as the Lagrange polynomial, such that $P(g(a)) = f_{\mathcal{P}}(a)$. Let k be the maximum grade in a monomial from $P(g(a))$ where we take the variables $\{a_i\}$. Then $P(g(a)) = \hat{P}(\hat{a})$ for some polynomial \hat{P} of grade 1 and $\hat{a} \in \mathcal{A}(k)$. Although \hat{P} is a regression model, it takes only zero or one values in the patterns $\hat{a} \in \mathcal{A}(k)$ and therefore can be taken as a linear classification model. \square

We present an example with Table 3, the first two columns with the class corresponding to a dataset distribution \mathcal{P} for an XOR function, which is not linearly separable. However, the 2–monomial expansion \mathcal{P}^2 is a linearly separable dataset distribution.

Table 3. XOR function under 2-monomial expansion.

a_1	a_2	a_1^2	$a_1 a_2$	a_2^2	Class
1	1	1	1	1	0
1	0	1	0	0	1
0	1	0	0	1	1
0	0	0	0	0	0

Definition 7. Let $\{\mathcal{P}_i\}$ be a sequence of dataset distributions, if \mathcal{P}_{i+1} is an extension of \mathcal{P}_i for all i , then $\{\mathcal{P}_i\}$ is a progressive sequence of dataset distributions.

This definition seeks to formalize the notion of a feature construction method that is applied iteratively, producing an unbounded quantity of new features. For example, if we construct each k -monomial extension of \mathcal{P} , such that the features of \mathcal{P}^k have the same indices in \mathcal{P}^{k+1} , then $\{\mathcal{P}^i\}$ is a progressive sequence of dataset distributions.

Definition 8. We say that a feature construction method is linearly asymptotic if from all dataset distribution \mathcal{P} over $\mathcal{A} \times \{0, 1\}$, feature construction methods produce a progressive sequence of dataset distributions $\{\mathcal{P}_i\}$, such that there is some k and a linear classifier that can compute $f_{\mathcal{P}^k}$ from \mathcal{P}^k .

Finally, we present a desirable property for any feature construction method. This property is equivalent to a feature construction method never getting stuck in patterns that are not linearly separable. Proving that a feature construction method is linearly asymptotic represents a formal validation of the method. For example, by Theorem 2, we conclude that the k –monomial construction method is linearly asymptotic.

Note that this desired property is similar to the kernel trick exploited by SVM models, where the data are mapped to a larger-dimensional space, such that a low-capacity classifier can separate the classes [42].

5. Experimental Results

In this section, we present the experimental results. We analyze the accuracy under the application of classification algorithms on pre-processed real and artificial datasets with their k -monomial extensions. The classification algorithms used are Naive Bayes, logistic regression, KNN, PART, JRIP, J48, and random forest. The classifiers mentioned were executed using the Waikato Environment for Knowledge Analysis (Weka) software [43].

5.1. Datasets from Real Classification Problems

The real data correspond to the Speaker Accent Recognition dataset [44], Algerian Forest Fires dataset [45], Banknote Authentication dataset [46], User Knowledge Modeling dataset [47], Glass Identification dataset [48], Wine Quality dataset [49], Somerville Happiness Survey dataset [50], Melanoma dataset, and Pima Indians Diabetes dataset [51]. As the experimental analysis is limited to binary classification problems, we took only the instances that belong to one of the two majority classes in the case of the Speaker Accent Recognition dataset, User Knowledge Modeling dataset, Glass Identification dataset, and Wine Quality dataset.

Before the analysis, we applied the k-monomial extension for $k = 2$ and 3 in the datasets obtaining two new datasets per original dataset. Finally, we applied a normalization

$$f(a_i) = \frac{(a_i - \inf A_i)}{(\sup A_i - \inf A_i)} \tag{5}$$

on all datasets and features A_i , where $a_i \in A_i$. Table A1 shows more details about the datasets and their k-monomial extensions.

5.2. Datasets from Artificial Classification Problems

The synthetic datasets are generated according to five rules that organize the datasets into five corresponding families. We first generate n features with r possible values for each dataset. The value of each feature given in an instance is generated from the ceiling function applied on a value x with uniform distribution in the interval $[0, r]$. For each rule, four datasets are generated with the following characteristics: (1) 2 features and 50 possible values; (2) 3 features and 30 possible values; (3) 4 features and 10 possible values; (4) 4 features and 5 possible values. The five binary rules for assigning classes to each instance are described below:

- The first rule assigns the category TRUE if the function,

$$Y_n^r : \{1, 2, \dots, r\}^n \rightarrow \{TRUE, FALSE\},$$

is greater than zero and otherwise assigns the category FALSE. The function Y_n^r is defined as:

$$Y_n^r(a) = \cos\left(\frac{(\sum_{i=1}^n a_i)\pi}{(r-1)n}\right). \tag{6}$$

- The second rule assigns the category TRUE if the function,

$$\Phi_n^r : \{1, 2, \dots, r\}^n \rightarrow \{TRUE, FALSE\},$$

is greater than zero, and otherwise assigns the category FALSE. The function Φ_n^r is defined as:

$$\Phi_n^r(a) = \prod_{i=1}^n \cos\left(\frac{a_i\pi}{r-1}\right). \tag{7}$$

- The third rule assigns the category TRUE if the function,

$$\Psi_n^r : \{1, 2, \dots, r\}^n \rightarrow \{TRUE, FALSE\},$$

is greater than zero, and otherwise it assigns the category FALSE. The function Ψ_n^r is defined as:

$$\Psi_n^r(a) = \left(\prod_{i=1}^n (a_i + 1)\right) - \left(\frac{r}{2}\right)^n. \tag{8}$$

- The fourth rule assigns the category TRUE if the function,

$$\Omega_n^r : \{1, 2, \dots, r\}^n \rightarrow \{TRUE, FALSE\},$$

is greater than zero, and otherwise assigns the category FALSE. The function Ω_n^r is defined as:

$$\Omega_n^r(a) = \left(\sum_{i=1}^n \left(a_i - \frac{r-1}{2} \right)^2 \right) - \left(\frac{\sqrt{n}(r-1)}{3} \right)^2. \tag{9}$$

- The fifth rule assigns the category TRUE if the function,

$$\Gamma_n^r : \{1, 2, \dots, r\}^n \rightarrow \{TRUE, FALSE\},$$

is greater than zero, and otherwise assigns the category FALSE. The function Γ_n^r is defined as:

$$\Gamma_n^r(a) = \left(\sum_{i=1}^n a_i \right) - \frac{nr}{2}. \tag{10}$$

Before the analysis, we applied the k-monomial extension for k = 2, 3, 4, and 5 in the datasets obtaining four new datasets per original dataset. Finally, we applied the normalization

$$f(a_i) = \frac{(a_i - \inf A_i)}{(\sup A_i - \inf A_i)} \tag{11}$$

on all datasets and features A_i , where $a_i \in A_i$. Table A6 shows more details about the datasets and their k-monomial extensions.

5.3. Analysis from the Real Datasets

In this subsection, we present the results corresponding to the real datasets. For the real datasets we have graphics like Figure 2 for the Speaker Accent Recognition dataset, that show the true positive, true negative, false positive, and false negative of the classification algorithms on each dataset, and their k-monomial extensions (Figures A1–A8, corresponding to the rest of the datasets are in the appendix). The values are calculated using 10-fold cross validation. For each algorithm, three joined bars are presented, showing the configuration of the confusion matrix. From left to right, the first bar corresponds to the original dataset, the second corresponds to the 2-monomial extension, and the last one corresponds to the 3-monomial extension. We represent the confusion matrix to show that the criteria for evaluating improvements in classification are adequate for these examples. We can see that there is little difference between the values of the original dataset and the k-monomial extensions most of the time. However, there are a few cases where the original dataset presents a significantly better accuracy, such as the naive Bayes classifier in Figure A1 and the J48 classifier in Figure A2. However, there are some cases where some k-monomial extension presents some accuracy slightly higher than the original dataset.

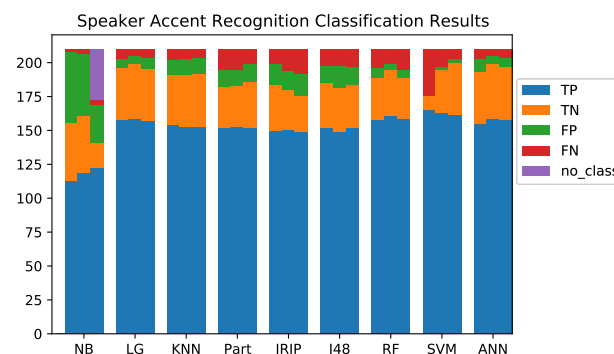


Figure 2. Graph corresponding to the Speaker Accent Recognition dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives and in purple are unclassified instances.

5.4. Analysis from the Artificial Datasets

In this subsection, we present the results corresponding to the artificial datasets. For the synthetic datasets we present results like Table 4 (for the first family of datasets) that shows the accuracy of the classification algorithms on each dataset and their k-monomial extensions (Tables A2–A5, corresponding to the rest of the families of datasets are in the Appendix A). The values are calculated using 10-fold cross-validation. Each dataset has a column indexed by “n-r”, where *n* is the number of features, and *r* is the cardinality of the features. For each dataset and algorithm, the original accuracy corresponds to the original dataset accuracy. Best accuracy corresponds to the highest precision between the k-monomial extensions, and grade corresponds to the *k* for which the k-monomial extensions reach the highest precision. In all families of datasets, we can see that the k-monomial extensions tend to have better accuracy than the original datasets. However, there are cases where the original dataset has more accuracy, but without exceeding 5%. We can also observe that the 5-monomial extension is common, as the case with greater accuracy. Notice that the 5-monomial extension is the dataset with a larger subset of redundant features.

Table 4. Results of artificial data from Family 1, where we only show the accuracy for the best values of *k*.

		Family 1			
		2-50	3-30	4-10	5-5
Naive Bayes	Original Accuracy	46.80	50.00	54.33	62.50
	Best Accuracy	56.00	51.60	53.67	65.00
	Grade	3	2	2	2
Logistic Regression	Original Accuracy	47.80	52.00	53.33	64.00
	Best Accuracy	99.20	56.20	54.67	65.00
	Grade	5	5	3	2
KNN	Original Accuracy	94.80	82.20	53.00	61.50
	Best Accuracy	94.80	84.40	53.67	64.5
	Grade	3	3-4-5	3	3-4
Rules PART	Original Accuracy	54.60	52.40	52.00	64.50
	Best Accuracy	96.20	63.20	52.67	64.50
	Grade	5	5	4-5	3-4
Rules JRip	Original Accuracy	86.20	53.00	54.33	59.50
	Best Accuracy	95.20	63.6	56.33	62.50
	Grade	3	5	4	3
Trees J48	Original Accuracy	54.60	52.00	51.33	68.00
	Best Accuracy	97.20	66.8	54.00	65.50
	Grade	3-4	5	5	4
Trees RF	Original Accuracy	93.40	68.60	51.67	65.50
	Best Accuracy	97.40	81.0	55.67	63.00
	Grade	4	3	2	5
SVM	Original Accuracy	50.80	49.60	55.67	64.00
	Best Accuracy	70.40	53.80	58.67	65.50
	Grade	4	2	3	3
ANN	Original Accuracy	90.00	51.20	50.67	59.50
	Best Accuracy	98.20	87.00	53.67	65.00
	Grade	4	3	4	2

6. Discussion

This is not the first work that relates features to data-complexity. The quotient between the number of instances and the number of features (known as the T2 measure) has been studied as a measure of data complexity [40]. However, T2 is independent of the notion of complexity in this work, since we can define linearly separable datasets in all ranges of T2. There are also applications of complexity measures for the feature selection problem, but applying a mainly experimental analysis [52–55].

The concept of a redundant set of features is based on the relevant feature definition of John et al. [56]. There are several other definitions for redundancy or redundant features. However, these definitions are more oriented to applications than a theoretical analysis of redundancy and its effects [57–64].

Our theoretical results show that many redundant features can reduce the complexity of the data. This result is interpreted in that a feature can provide representativeness without providing extra information, as seen in the example in Section 3. It can also be interpreted that redundant features are capable of increasing the capacity of the model.

Our experimental results reinforce the evidence that redundancy itself is not necessarily detrimental. The real and synthetic datasets showed that extended datasets with many redundant features constructed as monomials could achieve higher accuracy. However, higher accuracy was more pronounced in synthetic datasets. The synthetic datasets applied did not have noise and had few dimensions, which are the main differences to the real datasets studied.

Usually, redundant features before preprocessing entail a greater complexity of the algorithm than the classifier induces. The reason is that the classifier cannot find the optimal (global) rule, because the search space increases exponentially. Therefore, it returns a local optimum. Due to this increased search space, as we increase the features, the problem increases the difficulty and, tends to be classifiers with poorer performance. However, this fact occurs because those initial features do not add enough expressiveness. Therefore, features obtained from suitable construction methods cannot be equally treated in the same way as an initial feature.

Finally, the increase or decrease of features implies an increase or decrease of parameters in the model, respectively. Therefore, the choice of features can induce overfitting or underfitting. However, these learning problems are not commonly studied in the development of feature selection methods. Therefore, the criteria for selecting features should consider the information provided by each feature and the representativeness provided by the features. Furthermore, in the same way that there are regularization methods to avoid overfitting by the internal parameters of the model, regularization methods could be developed against the excess of features.

7. Conclusions

The main finding of this work is that attributes that are redundant from an information viewpoint indeed reduce under-fitting. Theoretical and experimental evidence is provided for this finding. However, these results are limited to binary classification problems with numerical attributes. Therefore, continuations of this work can be extended on the following points:

- Extension of the analysis on multi-class classification problems and with a significant proportion of categorical attributes.
- Extension of the analysis on regression problems.
- Extension of the analysis on models with a large number of parameters, where the phenomenon of under-fitting is unlikely, such as deep learning models.

Author Contributions: Conceptualization, S.A.G.; Formal analysis, S.A.G.; Investigation, S.A.G. and M.G.-T.; Project administration, J.L.V.N.; Software, L.R.B.P. and D.N.L.C.; Validation, J.L.V.N.; Visualization, J.C.M.R.; Writing—original draft, S.A.G.; Writing—review and editing, J.L.V.N., M.G.-T., J.F., D.P.P.-R., L.S.R. and F.G.-V. All authors contributed equally to this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CONACYT-Paraguay grant number PINV18-1199.

Data Availability Statement: The Algerian Forest Fires, Banknote Authentication, User Knowledge Modeling, Glass Identification, Somerville Happiness Survey and Wine quality data sets are available at <https://archive.ics.uci.edu/ml/index.php>, accessed on 24 September 2021. The Pima Indians Diabetes data set is available at <https://www.kaggle.com/uciml/pima-indians-diabetes-database>, accessed on 24 September 2021. The artificial data-sets are available at <https://drive.google.com/drive/folders/1RW4EAR4ZxP8ZHCW1ErOMCAg24EXEgHFB?usp=sharing>, accessed on 4 November 2021.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

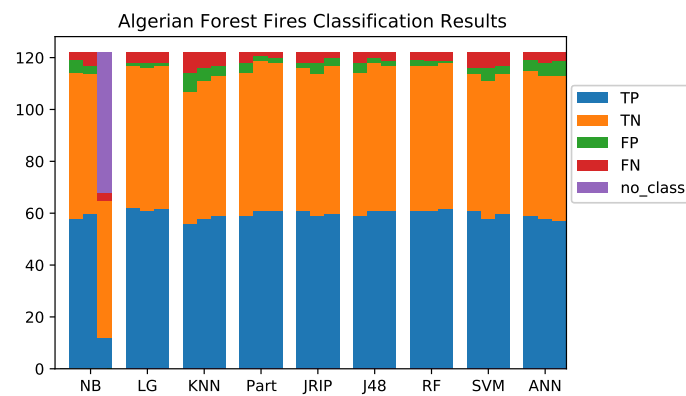


Figure A1. Graph corresponding to the Algerian Forest Fires dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives, and in purple are unclassified instances.

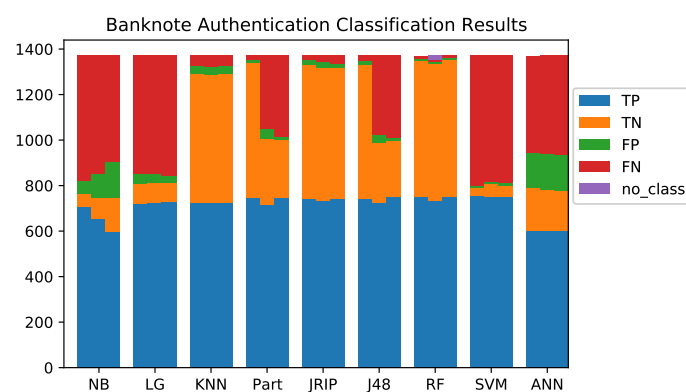


Figure A2. Graph corresponding to the Banknote Authentication dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives, and in purple are unclassified instances.

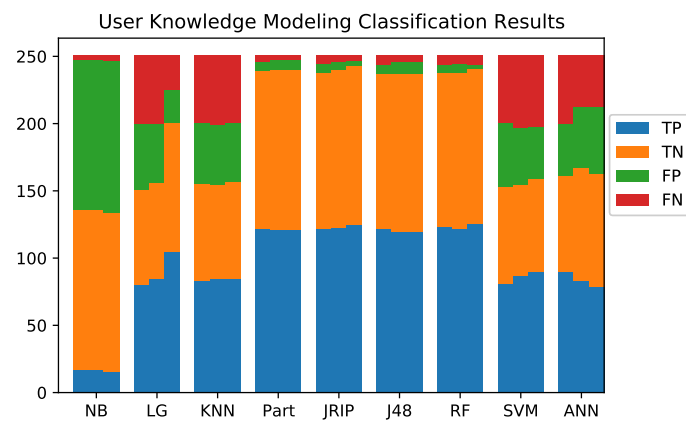


Figure A3. Graph corresponding to the User Knowledge Modeling dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives, and in purple are unclassified instances.

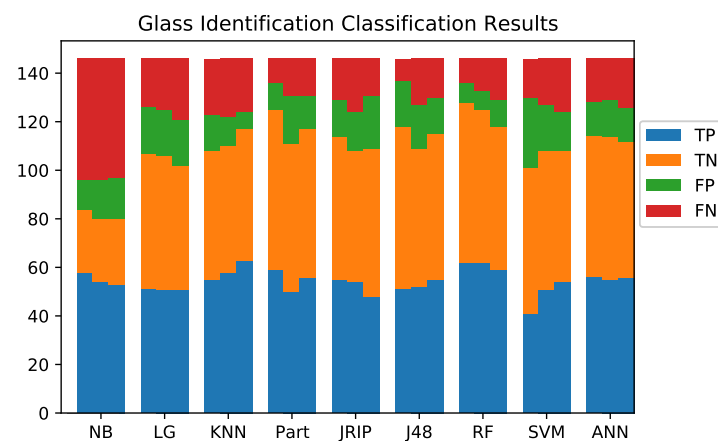


Figure A4. Graph corresponding to the Glass Identification dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives, and in purple are unclassified instances.

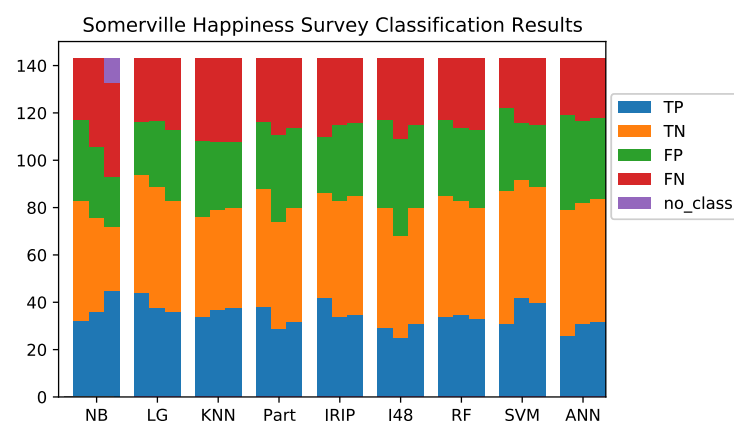


Figure A5. Graph corresponding to the Somerville Happiness Survey dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives, and in purple are unclassified instances.

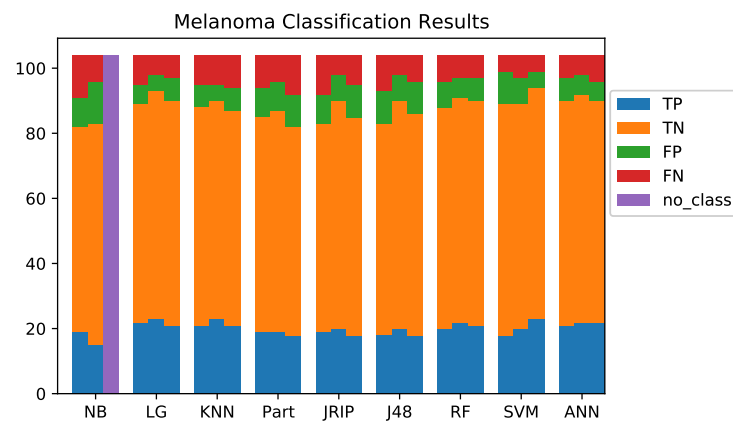


Figure A6. Graph corresponding to the Melanoma dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives, and in purple are unclassified instances.

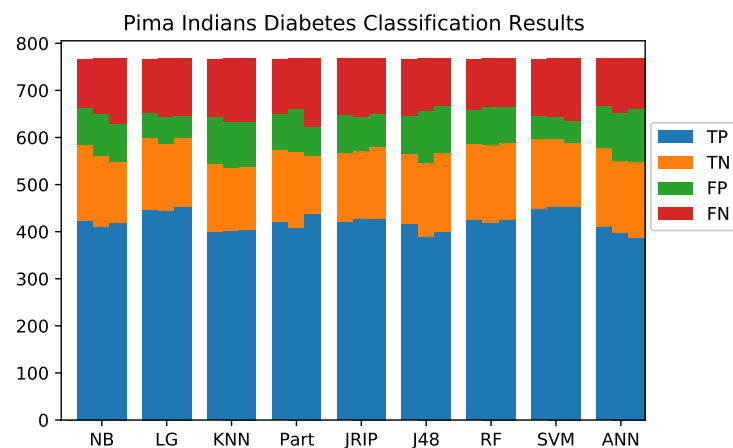


Figure A7. Graph corresponding to the Pima Indians Diabetes dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives, and in purple are unclassified instances.

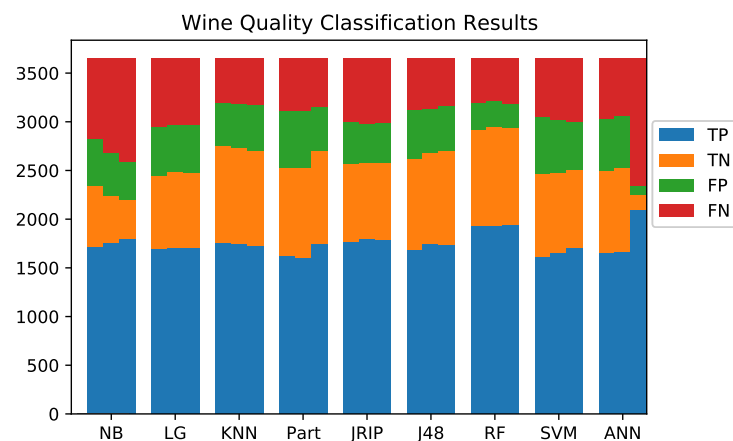


Figure A8. Graph corresponding to the Wine Quality dataset. In blue are true positives, in orange are true negatives, in green are false positives, in red are false negatives, and in purple are unclassified instances.

Table A1. Basic information about the real datasets. The column “Instances” denotes the number of entries in the dataset. The column “Original” denotes the number of features in the original dataset. The columns “2-Mon. Ext.” and “3-Mon. Ext.” denote the number of features in the 2-monomial extension and 3-monomial extension, respectively.

Data-Set	Instances	Original	2-Mon. Ext.	3-Mon. Ext.
Speaker Accent Recognition	210	12	90	454
Algerian Forest Fires	122	13	104	559
Banknote Authentication	1371	4	14	34
User Knowledge Modeling	251	5	20	55
Glass Identification	146	9	54	219
Melanoma	104	19	209	1539
Pima Indians Diabetes	767	8	44	164
Somerville Happiness Survey	143	6	27	83
Wine Quality	3655	11	77	363

Table A2. Table corresponding to results of artificial data from Family 2.

		Family 2			
		2-50	3-30	4-10	5-5
Naive Bayes	Original Accuracy	49.67	52.30	48.00	56.00
	Best Accuracy	71.33	57.10	49.00	53.33
	Grade	5	5	4	3-4
Logistic Regression	Original Accuracy	45.33	51.50	48.80	51.33
	Best Accuracy	99.00	99.20	91.80	56.0
	Grade	3	3	4	5
KNN	Original Accuracy	98.00	91.60	85.40	59.33
	Best Accuracy	97.33	91.10	80.20	62.67
	Grade	2-3-4	2	2	4
Rules PART	Original Accuracy	53.67	51.70	49.20	52.67
	Best Accuracy	98.33	95.60	69.40	53.33
	Grade	5	3	4	2
Rules JRip	Original Accuracy	99.33	99.70	61.20	52.00
	Best Accuracy	97.33	95.80	62.00	53.33
	Grade	5	2	3	2-3-4
Trees J48	Original Accuracy	53.67	53.30	49.20	53.33
	Best Accuracy	98.67	98.40	71.20	56.00
	Grade	2-3-5	5	5	2
Trees RF	Original Accuracy	99.33	99.80	80.80	49.33
	Best Accuracy	99.00	99.80	79.60	47.33
	Grade	3	2	3	2
SVM	Original Accuracy	53.67	52.10	45.60	54.67
	Best Accuracy	92.00	78.90	55.40	56.00
	Grade	5	5	5	3
ANN	Original Accuracy	81.67	65.90	65.60	53.33
	Best Accuracy	96.00	93.10	77.20	66.00
	Grade	3-4-5	4	3	3

Table A3. Table corresponding to results of artificial data from Family 3.

		Family 3			
		2-50	3-30	4-10	5-5
Naive Bayes	Original Accuracy	93.00	91.33	90.17	89.33
	Best Accuracy	98.00	91.67	88.33	89.00
	Grade	2	5	5	4
Logistic Regression	Original Accuracy	93.00	91.67	88.67	94.67
	Best Accuracy	97.00	98.83	96.50	96.33
	Grade	2-5	3-4	3	2
KNN	Original Accuracy	95.00	96.00	91.17	89.67
	Best Accuracy	97.00	96.33	89.83	87.33
	Grade	2-3-4-5	2	2	3
Rules PART	Original Accuracy	91.00	95.00	90.17	85.67
	Best Accuracy	99.00	99.50	97.67	92.00
	Grade	2-3-4-5	3-4-5	4-5	5
Rules JRip	Original Accuracy	94.00	93.33	87.83	78.33
	Best Accuracy	99.00	99.17	97.33	91.00
	Grade	2-3-4-5	3-4-5	5	5
Trees J48	Original Accuracy	90.00	94.17	90.33	83.00
	Best Accuracy	99.00	99.50	97.17	92.33
	Grade	2-3-4-5	3-4-5	5	4
Trees RF	Original Accuracy	94.00	96.67	93.33	88.33
	Best Accuracy	100.00	99.83	98.50	95.67
	Grade	2-3-4-5	5	5	5
SVM	Original Accuracy	93.00	91.33	89.17	92.67
	Best Accuracy	96.00	97.17	95.67	94.67
	Grade	5	4-5	5	5
ANN	Original Accuracy	96.00	93.00	94.33	93.00
	Best Accuracy	98.00	98.67	97.00	97.00
	Grade	2	5	4	2

Table A4. Table corresponding to results of artificial data from Family 4.

		Family 4			
		2-50	3-30	4-10	5-5
Naive Bayes	Original Accuracy	82.00	77.60	82.00	77.33
	Best Accuracy	76.00	80.10	75.00	77.67
	Grade	2-3-4-5	2	2	2
Logistic Regression	Original Accuracy	68.00	69.80	55.00	62.67
	Best Accuracy	98.00	99.40	95.00	96.67
	Grade	2-3	2	2	2

Table A4. Cont.

		Family 4			
		2-50	3-30	4-10	5-5
KNN	Original Accuracy	82.00	91.00	70.00	79.33
	Best Accuracy	82.00	91.40	71.00	75.67
	Grade	2-3-4-5	2	2	3
Rules PART	Original Accuracy	80.00	89.30	75.00	88.33
	Best Accuracy	82.00	91.50	77.00	90.00
	Grade	4	3	2-4	2
Rules JRip	Original Accuracy	72.00	88.40	62.00	83.67
	Best Accuracy	72.00	90.30	66.00	80.67
	Grade	2-3-5	3	2-3	5
Trees J48	Original Accuracy	78.00	91.00	62.00	85.33
	Best Accuracy	80.00	90.90	75.00	87.33
	Grade	3-4	2-5	3	2
Trees RF	Original Accuracy	76.00	93.60	73.00	88.33
	Best Accuracy	78.00	94.10	77.00	89.67
	Grade	2	4	3-4	3
SVM	Original Accuracy	66.00	69.80	61.00	64.67
	Best Accuracy	82.00	94.40	73.00	94.33
	Grade	4	5	4-5	4
ANN	Original Accuracy	84.00	75.90	66.00	69.67
	Best Accuracy	96.00	97.70	90.00	93.33
	Grade	3	2	2	2

Table A5. Table corresponding to results of artificial data from Family 5.

		Family 5			
		2-50	3-30	4-10	5-5
Naive Bayes	Original Accuracy	95.00	94.00	92.00	90.00
	Best Accuracy	98.00	98.00	94.25	92.00
	Grade	2-3-4	5	2	2
Logistic Regression	Original Accuracy	99.50	100.00	100.00	100.00
	Best Accuracy	99.50	100.00	99.75	93.00
	Grade	3	2	3	2
KNN	Original Accuracy	97.00	94.50	96.25	84.00
	Best Accuracy	97.00	94.50	96.50	85.00
	Grade	2-3-4-5	2	2	4-5
Rules PART	Original Accuracy	91.00	84.50	90.25	80.00
	Best Accuracy	97.00	95.00	94.50	88.00
	Grade	3	4	5	2

Table A5. Cont.

		Family 5			
		2-50	3-30	4-10	5-5
Rules JRip	Original Accuracy	92.50	82.50	90.25	78.00
	Best Accuracy	96.50	93.50	93.50	90.00
	Grade	2	3	4	4
Trees J48	Original Accuracy	90.50	87.00	89.25	79.00
	Best Accuracy	95.50	94.00	93.50	88.00
	Grade	3	2	4	3
Trees RF	Original Accuracy	94.50	90.00	93.75	85.00
	Best Accuracy	98.00	95.50	96.25	93.00
	Grade	3-4-5	4-5	3	5
SVM	Original Accuracy	96.00	96.50	96.50	92.00
	Best Accuracy	98.00	96.50	99.25	100.00
	Grade	2-5	2	2	4-5
ANN	Original Accuracy	100.00	100.00	100.00	100.00
	Best Accuracy	100.00	98.00	100.00	100.00
	Grade	3-4-5	2	2	2

Table A6. Basic information about the artificial datasets. The column “Family” denotes the corresponding family function. The column “Indices” denotes the number of features and their cardinality. The column “Instances” denotes the number of entries in the dataset. The column “Original” denotes the number of features in the original dataset. The columns “2-Mon. Ext.,” “3-Mon. Ext.,” “4-Mon. Ext.,” and “5-Mon. Ext.,” denote the number of features in the 2-monomial extension, 3-monomial extension, 4-monomial extension, and 5-monomial extension, respectively.

Family	Indices	Instances	Original	2-Mon. Ext.	3-Mon. Ext.	4-Mon. Ext.	5-Mon. Ext.
1	2-50	500	2	5	9	14	20
1	3-30	500	3	9	19	34	55
1	4-10	300	4	14	34	69	125
1	5-5	200	5	20	55	125	251
2	2-50	300	2	5	9	14	20
2	3-30	1000	3	9	19	34	55
2	4-10	500	4	14	34	69	125
2	5-5	150	5	20	55	125	251
3	2-50	100	2	5	9	14	20
3	3-30	600	3	9	19	34	55
3	4-10	600	4	14	34	69	125
3	5-5	300	5	20	55	125	251
4	2-50	50	2	5	9	14	20
4	3-30	1000	3	9	19	34	55
4	4-10	100	4	14	34	69	125
4	5-5	300	5	20	55	125	251
5	2-50	200	2	5	9	14	20
5	3-30	200	3	9	19	34	55
5	4-10	400	4	14	34	69	125
5	5-5	100	5	20	55	125	251

References

1. Vergara, J.R.; Estévez, P.A. A review of feature selection methods based on mutual information. *Neural Comput. Appl.* **2013**, *24*, 175–186. [\[CrossRef\]](#)
2. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
3. Sondhi, P. Feature construction methods: A survey. *Sifaka. Cs. Uiu. Edu.* **2009**, *69*, 70–71.
4. Tang, J.; Alelyani, S.; Liu, H. Feature selection for classification: A review. *Data Classif. Algorithms Appl.* **2014**, *37*. [\[CrossRef\]](#)
5. Yang, C.H.; Chuang, L.Y.; Yang, C.H. IG-GA: A Hybrid Filter/Wrapper Method for Feature Selection of Microarray Data. *J. Med. Biol. Eng.* **2010**, *30*, 23–28.
6. Hsu, H.H.; Hsieh, C.W.; Lu, M.D. Hybrid feature selection by combining filters and wrappers. *Expert Syst. Appl.* **2011**, *38*, 8144–8150. [\[CrossRef\]](#)
7. Chandrashekar, G.; Sahin, F. A Survey on Feature Selection Methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [\[CrossRef\]](#)
8. Venkatesh, B.; Anuradha, J. A review of feature selection and its methods. *Cybern. Inf. Technol.* **2019**, *19*, 3–26. [\[CrossRef\]](#)
9. Pudil, P.; Novovičová, J.; Kittler, J. Floating search methods in feature selection. *Pattern Recognit. Lett.* **1994**, *15*, 1119–1125. [\[CrossRef\]](#)
10. Ferri, F.J.; Pudil, P.; Hatf, M.; Kittler, J. Comparative study of techniques for large-scale feature selection. In *Machine Intelligence and Pattern Recognition*; Elsevier: Amsterdam, The Netherlands, 1994; Volume 16, pp. 403–413.
11. Ghosh, K.K.; Ahmed, S.; Singh, P.K.; Geem, Z.W.; Sarkar, R. Improved binary sailfish optimizer based on adaptive β -hill climbing for feature selection. *IEEE Access* **2020**, *8*, 83548–83560. [\[CrossRef\]](#)
12. Yan, C.; Ma, J.; Luo, H.; Wang, J. A hybrid algorithm based on binary chemical reaction optimization and tabu search for feature selection of high-dimensional biomedical data. *Tsinghua Sci. Technol.* **2018**, *23*, 733–743. [\[CrossRef\]](#)
13. Sayed, S.; Nassef, M.; Badr, A.; Farag, I. A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets. *Expert Syst. Appl.* **2019**, *121*, 233–243. [\[CrossRef\]](#)
14. Jia, H.; Li, J.; Song, W.; Peng, X.; Lang, C.; Li, Y. Spotted hyena optimization algorithm with simulated annealing for feature selection. *IEEE Access* **2019**, *7*, 71943–71962. [\[CrossRef\]](#)
15. Mafarja, M.M.; Mirjalili, S. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [\[CrossRef\]](#)
16. Paniri, M.; Dowlatshahi, M.B.; Nezamabadi-pour, H. MLACO: A multi-label feature selection algorithm based on ant colony optimization. *Knowl.-Based Syst.* **2020**, *192*, 105285. [\[CrossRef\]](#)
17. Gharehchopogh, F.S.; Maleki, I.; Dizaji, Z.A. Chaotic vortex search algorithm: Metaheuristic algorithm for feature selection. *Evol. Intell.* **2021**, 1–32. [\[CrossRef\]](#)
18. Sakri, S.B.; Rashid, N.B.A.; Zain, Z.M. Particle swarm optimization feature selection for breast cancer recurrence prediction. *IEEE Access* **2018**, *6*, 29637–29647. [\[CrossRef\]](#)
19. Liu, H.; Motoda, H. *Feature Extraction, Construction and Selection: A Data Mining Perspective*; Springer Science & Business Media: New York, USA, 2012; Volume 453.
20. Mahanipour, A.; Nezamabadi-Pour, H.; Nikpour, B. Using fuzzy-rough set feature selection for feature construction based on genetic programming. In Proceedings of the 2018 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), Bam, Iran, 6–8 March 2018; pp. 1–6.
21. Neshatian, K.; Zhang, M.; Andrae, P. A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. *IEEE Trans. Evol. Comput.* **2012**, *16*, 645–661. [\[CrossRef\]](#)
22. Markovitch, S.; Rosenstein, D. Feature generation using general constructor functions. *Mach. Learn.* **2002**, *49*, 59–98. [\[CrossRef\]](#)
23. Fan, W.; Zhong, E.; Peng, J.; Verscheure, O.; Zhang, K.; Ren, J.; Yan, R.; Yang, Q. Generalized and heuristic-free feature construction for improved accuracy. In Proceedings of the 2010 SIAM International Conference on Data Mining, Columbus, OH, USA, 29 April–1 May 2010; pp. 629–640.
24. Ma, J.; Gao, X. A filter-based feature construction and feature selection approach for classification using Genetic Programming. *Knowl.-Based Syst.* **2020**, *196*, 105806. [\[CrossRef\]](#)
25. Tran, B.; Xue, B.; Zhang, M. Genetic programming for multiple-feature construction on high-dimensional classification. *Pattern Recognit.* **2019**, *93*, 404–417. [\[CrossRef\]](#)
26. Specia, L.; Srinivasan, A.; Ramakrishnan, G.; Nunes, M.d.G.V. Word sense disambiguation using inductive logic programming. In *Proceedings of the 16th International Conference, ILP 2006, Santiago de Compostela, Spain, 24–27 August 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 409–423.
27. Specia, L.; Srinivasan, A.; Joshi, S.; Ramakrishnan, G.; Nunes, M.d.G.V. An investigation into feature construction to assist word sense disambiguation. *Mach. Learn.* **2009**, *76*, 109–136. [\[CrossRef\]](#)
28. Roth, D.; Small, K. Interactive feature space construction using semantic information. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), Boulder, CO, USA, 4–5 June 2009; pp. 66–74.
29. Derczynski, L.; Chester, S. Generalised Brown clustering and roll-up feature generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
30. iwek, K.; Osowski, S. *Comparison of Methods of Feature Generation for Face Recognition*; University of West Bohemia: Pilsen, Czechia, 2013.

31. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification. A Wiley-Interscience Publication*, 2nd ed.; John Wiley & Sons, Inc.: New York, NY, USA, 2001.
32. Sutton, R.S.; Matheus, C.J. Learning polynomial functions by feature construction. In *Machine Learning Proceedings 1991*; Elsevier: San Mateo, CA, USA, 1991; pp. 208–212.
33. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, USA, 4–6 August 2001; Volume 3, pp. 41–46.
34. Wright, R.E. Logistic regression. In *Reading and Understanding Multivariate Statistics*; Grimm, L.G., Yarnold, P.R., Eds.; American Psychological Association: Washington, DC, USA, 1995; pp. 217–244.
35. Fix, E.; Hodges, J.L. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *Int. Stat. Rev./Revue Internationale de Statistique* **1989**, *57*, 238–247. [[CrossRef](#)]
36. Frank, E.; Witten, I.H. Generating Accurate Rule Sets Without Global Optimization. In Proceedings of the Fifteenth International Conference on Machine Learning, Madison, WI, USA, 24–27 July 1998; Shavlik, J., Ed.; Morgan Kaufmann: San Francisco, CA, USA, 1998; pp. 144–151, ISBN 978-1-55860-556-5.
37. Cohen, W.W. Fast Effective Rule Induction. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; Morgan Kaufmann: San Francisco, CA, USA, 1995; pp. 115–123, ISBN 978-1-55860-377-6.
38. Quinlan, R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann Publishers: San Mateo, CA, USA, 1993.
39. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
40. Ho, T.K.; Basu, M. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 289–300.
41. Blumer, A.; Ehrenfeucht, A.; Haussler, D.; Warmuth, M.K. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM* **1989**, *36*, 929–965. [[CrossRef](#)]
42. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [[CrossRef](#)]
43. Garner, S.R.; others. Weka: The waikato environment for knowledge analysis. In Proceedings of the New Zealand Computer Science Research Students Conference, Hamilton, New Zealand, 14–18 April 1995; Volume 1995, pp. 57–64.
44. Fokoue, E. Speaker Accent Recognition Data Set. UCI Machine Learning Repository, 2020. Available online: <https://archive.ics.uci.edu/ml/datasets/Speaker+Accent+Recognition> (accessed on 24 September 2021)
45. Abid, F.; Izeboudjen, N. Predicting Forest Fire in Algeria Using Data Mining Techniques: Case Study of the Decision Tree Algorithm. In Proceedings of the International Conference on Advanced Intelligent Systems for Sustainable Development, Tangier, Morocco, 12–14 July 2019; Springer: Cham, Switzerland, 2019; pp. 363–370.
46. Lohweg, V. Banknote authentication Data Set. UCI Machine Learning Repository, 2012. Available online: <https://archive.ics.uci.edu/ml/datasets/banknote+authentication> (accessed on 24 September 2021)
47. Kahraman, H.T.; Sagioglu, S.; Colak, I. The development of intuitive knowledge classifier and the modeling of domain dependent data. *Knowl.-Based Syst.* **2013**, *37*, 283–295. [[CrossRef](#)]
48. German, B. Glass Identification Data Set. UCI Machine Learning Repository, 1987. Available online: <https://archive.ics.uci.edu/ml/datasets/glass+identification> (accessed on 24 September 2021)
49. Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decis. Support Syst.* **2009**, *47*, 547–553. [[CrossRef](#)]
50. Koczkodaj, W.W. Somerville Happiness Survey Data Set. UCI Machine Learning Repository, 2018. Available online: <https://archive.ics.uci.edu/ml/datasets/Somerville+Happiness+Survey> (accessed on 24 September 2021)
51. Rossi, R.A.; Ahmed, N.K. The Network Data Repository with Interactive Graph Analytics and Visualization. In Proceedings of the AAAI15: Twenty-Ninth Conference on Artificial Intelligence, Austin, TX, USA, 25–30 July 2015.
52. Seijo-Pardo, B.; Bolón-Canedo, V.; Alonso-Betanzos, A. Using data complexity measures for thresholding in feature selection rankers. In Proceedings of the Conference of the Spanish Association for Artificial Intelligence, Salamanca, Spain, 14–16 September 2016; Springer: Cham, Switzerland, 2016; pp. 121–131.
53. Dom, B.; Niblack, W.; Sheinvald, J. Feature selection with stochastic complexity. In Proceedings of the 1989 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 4–8 June 1989; pp. 241–242.
54. Bolón-Canedo, V.; Sánchez-Maróño, N.; Alonso-Betanzos, A. A distributed feature selection approach based on a complexity measure. In Proceedings of the International Work-Conference on Artificial Neural Networks, Palma de Mallorca, Spain, 10–12 June 2015; Springer: Cham, Switzerland, 2015; pp. 15–28.
55. Okimoto, L.C.; Lorena, A.C. Data complexity measures in feature selection. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
56. John, G.H.; Kohavi, R.; Pflieger, K. Irrelevant features and the subset selection problem. In Proceedings of the Eleventh International Conference on Machine Learning, New Brunswick, NJ, USA, 10–13 July 1994; pp. 121–129.
57. Ding, C.; Peng, H. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol.* **2005**, *3*, 185–205. [[CrossRef](#)]
58. Yu, L.; Liu, H. Efficient feature selection via analysis of relevance and redundancy. *J. Mach. Learn. Res.* **2004**, *5*, 1205–1224.
59. Gao, W.; Hu, L.; Zhang, P. Feature redundancy term variation for mutual information-based feature selection. *Appl. Intell.* **2020**, *50*, 1272–1288. [[CrossRef](#)]

60. Zhou, T.; Zhang, C.; Gong, C.; Bhaskar, H.; Yang, J. Multiview latent space learning with feature redundancy minimization. *IEEE Trans. Cybern.* **2018**, *50*, 1655–1668. [[CrossRef](#)]
61. Cheng, G.; Qin, Z.; Feng, C.; Wang, Y.; Li, F. Conditional Mutual Information-Based Feature Selection Analyzing for Synergy and Redundancy. *ETRI J.* **2011**, *33*, 210–218. [[CrossRef](#)]
62. Zhao, Z.; Wang, L.; Liu, H. Efficient spectral feature selection with minimum redundancy. In Proceedings of the AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010; Volume 24.
63. Tabakhi, S.; Moradi, P. Relevance–redundancy feature selection based on ant colony optimization. *Pattern Recognit.* **2015**, *48*, 2798–2811. [[CrossRef](#)]
64. Wang, M.; Tao, X.; Han, F. A New Method for Redundancy Analysis in Feature Selection. In Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence, Sanya, China, 24–26 December 2020; pp. 1–5.