Feature Selection with Association Rules For Regression Problems

Zvi Berger (206126575), Ofir Nassimi (209374297)

Submitted as final project report for the Tabular Data Science course, BIU, Semester A, 2022

Abstract

Feature selection is one of the key issues in pattern recognition. The quality of the feature selection has a direct impact on the classification accuracy and generalization performance of the classifier. In order to reduce the size of the feature subset and improve the efficiency of the algorithm without reducing the accuracy, this paper proposes a feature selection algorithm based on association rules.

1 Introduction

Many times, data sets for analysis contain hundreds of attributes, which may be irrelevant to the mining task or redundant. Attribute subset selection or Feature selection is a technique to extract closely related features and remove irrelevant or useless features according to an objective function.

John et al. [9] considers feature selection to be a process of reducing feature dimensions without reducing classification accuracy. Koller et al. [4] defines feature selection to select as small a feature subset as possible, while ensuring that the result class distribution is as similar as possible to the original data class distribution. Dash et al. [18] gave a comprehensive overview of the feature selection problem in the field of data mining, and gave the basic framework of feature selection. From the evaluation cri- teria of feature sets, feature selection methods can be divided into the following three categories: Embedded, Filter and Wrapper [8]. In the embedded structure, the feature selection algorithm itself is embedded as a component in the classification algorithm, the most typical is the decision tree algorithm. The evaluation criteria for filter feature selection are directly obtained from the data set, independent of the classification algorithm, and are suitable for large-scale data sets. Kira et al. [15] proposed an effective feature selection algorithm, Relief algorithm. The wrapped feature selection algorithm has higher accuracy than the filter feature selection algorithm, but the algorithm is less efficient.

The aim of Feature selection is to minimize the number of features such that the probability distribution of the resulting data classes is near to the original distribution of all the features [11]. An exhaustive search for the optimal subset of attributes can be prohibitively expensive, especially as total number of records (n) and the number of data classes increase. Association rule mining, one of the most important and well researched techniques of data mining. This technique is utilized in our work with reduced data set related to the desired class label and with reduced features. There are many reasons for subset selection of the features instead of all the features [22]. To measure a diminished set of features is cheaper, faster with increased accuracy by exclusion of irrelevant features. Differentiating relevant and irrelevant features, gives a proper insight about the nature of prediction problem.

For feature selection various heuristic methods used are stepwise forward selection, stepwise backward elimination, combined forward selection and backward elimination, random generation and decision tree induction. Stepwise forward selection is a feature selection method which starts with an empty set of attributes, best of the original attributes is found and added entirely after a single consideration of its usefulness. The pitfalls of this method include a high susceptibility to getting trapped by local optima, and a one track process that easily discards a feature entirely after a single consideration of its usefulness. Variations of this method is found in [20] [14] [1].

2 Related Work

For feature selection, there are many methods in the literature covering a wide range from filtering to wrapping approaches [17, 10]. In the filter approach, the goodness of an attribute or set of attributes is estimated by using only intrinsic properties of the data, while in the wrapper approach, the merit of a given candidate subset is obtained by learning and evaluating a classifier using only the variables included in the proposed subset [21]. Principal component analysis (PCA) and linear discriminant analysis (LDA) are the popular feature selection methods to reduce size [16]. In the recent years, many methods have been used for feature selection; particularly artificial intelligence, feature conversion methods and statistical methods: boosting feature selection for neural network based regression [2], filter model for feature subset selecting based on genetic algorithm [7], application of ant colony algorithm for feature selection [19], feature selection using particle swarm optimization [3], a discrete particle swarm optimization method for feature selection [24], feature selection by Weighted-SNR for cancer microarray data classification [12], Bhattacharyya space for feature selection [23], subspace based feature selection method [10], support vector-based feature selection using fisher's linear discriminant and support vector machine [6], HMM (Hidden Markov Models) based feature space transform for voice pathology detection [13] have been used.

3 Association Rule Mining

Let I = i1, i2, i3,..., id be the set of all items in a market basket data and T = t1,t2,t3,...,tn be the set of all transactions. Each transaction ti contains a subset of items chosen from Item set I. A collection of zero or more items is termed an item set. Support count is an important property of an item set. Support count refers to the number of transactions that contain a particular item set. Mathematically, the support count, $\sigma(X)$, for an item set X can be given as follows:

$$\sigma(X) = |\{t_i | X \subseteq t_i, t_i \in T_i\}|$$

An association rule is an implication expression of the form A → B, where A and B are disjoint item sets, i.e., $A \cap B = \emptyset$. There are two important basic measures for association rules, minimum support and confidence. Generally minimum support and confidence are predefined by user/analyst so that the rules which are not so interesting or not useful can be deleted. Support is the total count of number of transactions where all items in A and B are together. Confidence determines how frequently items in B appear in transactions that contain A. The formal definitions of these metrics are given below,

$$Support(A \rightarrow B) = \sigma(A \wedge B)$$
$$Confidence(A \rightarrow B) = \sigma(A \wedge B)/\sigma(A)$$

Apriori algorithm is used to find the frequent item sets [5].

4 Dataset

We test our approach on 4 different datasets. House Price, Diamond Price, NBA Rookie and Avocado Price.

4.1 House Price

This huge dataset contains 80 features about house prices including price, type of road access, type of roof, roof material, height of the basement, number of kitchens, kitchen quality, number of fireplaces, year garage was built, size of garage in car capacity, wood deck area in square feet, pool quality, year sold, condition of sale and more. Out goal is to try to predict the price of the houses by all of it's features or even by just part of them, which means less than 80. You can find this dataset at Kaggle.

Dimensions: (1461, 80).
Targets: Price.

4.2 Diamond Price

This dataset contains the prices and other attributes of almost 54,000 diamonds. You can find this data at Kaggle. This dataset contain: price price in US dollars ($326–$18,823), carat weight of the diamond (0.2–5.01), cut quality of the cut (Fair, Good, Very Good, Premium, Ideal), color diamond colour, from J (worst) to D (best), clarity a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best)), x length in mm (0–10.74), y width in mm (0–58.9), z depth in mm (0–31.8), depth total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43–79), table width of top of diamond relative to widest point (43–95). Our predict target is diamond price.

Dimensions: (53940, 10).
Targets: ‘carat’ or ‘price’.

4.3 NBA Rookie

This dataset of NBA Rookie stats is used in order to practice classification problem - will a player last 5 years or more in the league. This dataset contains 21 features about rookie players including their names, amount of games played, minutes played, regular goals made as well as attempts, 3 point goals made and attempt and the percentage of success etc. By all of these features we attempt to predict the career longevity of the players - 0 if the career lasts less than 5 years or 1 if more than that. You can find this data at data world.

Dimensions: (8128, 12).
Target: $TARGET\_5Yrs$.

4.4 Avocado Price

This dataset contains the prices and more information about many avocados from different cities and states in United States in the years 2015 and 2016. You can find this data at Kaggle. With the help of this dataset we want to predict the average price of an avocado by it’s information. This dataset contains: average price in US dollars ($0.49–$2.79), date (04.01.2015–22.05.2016), total volume - total number of avocados sold (84.56-52.2m), PLU 4046 sold - total number of avocados with PLU 4046 sold (0-18.9m), PLU 4225 sold - total number of avocados with PLU 4225 sold (0-20.4m), PLU 4770 sold - total number of avocados with PLU 4770 sold (0-2.5m), total bags sold (0-12.6m), small bags sold (0-9m), large bags sold (0-3.3m), XL bags sold (0-241.6k), type of the avocado (Conventional, Organic), year (2015-2016) and geography (cities in US).

Dimensions: (7819, 13).
Target: ’average_price’.

5 Solution

5.1 ARFS

In order to reduce the size of the feature subset and improve the efficiency of the feature selection algorithm without reducing the classification accuracy, we propose an ARFS (Association Rules Features Selection) algorithm. The ARFS algorithm first calculate the support, confidence and lift between the feature and the predict feature and use the values of each feature to evaluate the correlation between the feature and the predict feature. The features are then sorted by the size of their correlation weights, and an ordered sequence of features with a large correlation between feature and predict feature is obtained. Then we choose top k features with highest correlation. The pseudo code of the algorithm is described as follows.

Algorithm 1 ARFS

```
  Input Dataset d with n feature.
  Output Dataset d with k feature.
$support \leftarrow 0.01$
$confidence \leftarrow 0.6$
$k \leftarrow 7$
$rules \leftarrow Apriori(d, confidence, confidence)$
$rules \leftarrow$ sorted rules by lift
$features \leftarrow \{\}$
for $i = 0,\ i < len(rules)\ and\ len(features) < k,\ i{+}{+}$ do
    $features.append(rules[i])$
end for
predict with xgboost
```

5.2 WebSite

We built a website tool to present our approach and test her on differences datasets. The site contains 7 pages. The first 6 of them depend on each other. In order to continue to the next page, you need to pass the previous page first:

- Introduction page - an intro about our project, our goal and the ways to get there.
- Datasets page - shows in a table a dataset from all of our datasets that the system works on, including a table with other information such as mean and max value of every feature. You can choose to see every dataset from our veraity.
- Association rules page - finds all of the association rules in the dataset. In the page, you choose a dataset, a support and confidence thresholds and

click the "find association rules" button. A table with all rules found will be shown with their support, confidence, lift and other values.

- Association rules analysis page - this page first shows all of the rules that were found in the last page. After that, you choose one feature from all of the dataset's features and a table with all of the rules that include that feature will be shown.

- Feature selection page - in this page you choose the number of top features you want to detect in every concept (support, confidence, lift, distance to 1 of lift). A table with the top K features of each concept will be shown.

- Model comparison page - you choose a model to predict the features by from a wide veraity of models options. After clicking the "predict" button, the results will be shown on the screen with graphs that compares the predictions againts the true value.

- Test page - this page does not depend on the previous pages. Here you can do multiple tests on the datasets and perform in one click the whole website goal. First choose a dataset you want to test, then select the feature from all of the dataset's features and then choose the amount of tests you want. After clicking "Test", it will start and you will see on the screen the parts of the test that are done. Every part of the test includes different values of support threshold, confidence threshold and K (to find top-K). The test includes: 1. Finding association rules 2. Analysing the rules 3. Performing feature selection 4. Comparing the models. After the test ends, a folder with all of the information will be opened. This folder includes a statistics page, the test and train data and all of the information that is given in the other pages (graphs of each model for different K's, a csv file with all of the association rules that found, a csv file with the analysis of the rules and a csv file of the models comparisons). More information about this page is written in the page.

6 Experimental Results

We've experimented 4 datasets (the datasets are mentioned and explained above). On each dataset we've done 10 tests. Each test constructed by a few steps: first step is finding association rules with a support threshold on the dataset's features and analyzing them. The seconds step is finding the top K features of each concept (support, confidence, lift and distance of lift from 1). In the third and last step, a machine learning model is trained and tested 5 times and every time calculates the R-Square score and the mean error. Each time the model is trained by different data. First by the whole data and then just by the top K ones of each concept (only the top K features are taken into account). Those steps are repeated 24 times with different supports (0.01, 0.05, 0.1), different K's (3, 5, 7, 10) and different models (Linear Regression, XGBoost).

After doing the tests, each one returns, among other things, the R-Square score and the mean square error of each repetition of the steps.
In order to determine the best values of support, k and model that will lead us to the best predictions, we've combined all of the results of the 10 tests, calculated the average R-Square score of each K, support and model and chose the maximum one.

6.1 House Price

Testing the House Price dataset led to the results at table 1. The table contains a few of the total results. For example, training the XGBoost model with all of the features and support of 0.01 gave the max R-Square score of 0.9969 with a MSE of 161596438.6. You can also see the graph of XGBoost sorted by lift with a selection of top 10 features at Fig 1.

Model	Feature Select	Support	Features	R-Square Score	MSE
Linear Regression	All	0.01	All Features	0.75671042	1456150968
Linear Regression	10	0.01	Sort by Support	0.761391126	1395359557
Linear Regression	10	0.01	Sort by Lift	0.744984163	1484895791
XGBoost	All	0.01	All Features	0.971866602	161596438.6
XGBoost	10	0.01	Sort by Support	0.996971205	17344051.21
XGBoost	10	0.01	Sort by Confidence	0.996894898	17937676.62
XGBoost	7	0.01	Sort by Lift	0.990861027	54683371.86
XGBoost	10	0.01	Sort by Lift	0.997	16919821.64

Table 1: Sample Results on House Price Dataset

6.2 Diamond Price

Testing the diamonds dataset led to the following results: The maximum average R-Square score by using the Linear Regression model is 0.88 and it's co-responding MSE is 1783230 with training the model by using all of the features, support of 0.01 and with K = 3. Bigger K's (5, 7, 10) led to the same results if using all the features. But if only the features by support/confidence/lift are used, the R-Square error will be 0.85-0.87. The maximum average R-Square score by using XGBoost model is standing on 0.99 and it's co-responding MSE is 158669.7 with training the model by using all of the features, support of 0.01 and with K = 3. Bigger K's (5, 7, 10) led to the same results. The best results with a support of 0.1 are the same both for Linear Regression and XGBoost models. this support but using only features sorted by lift led to R-Square score - 0.78 and MSE 3442317. You can review a sample of the results in Table 2.

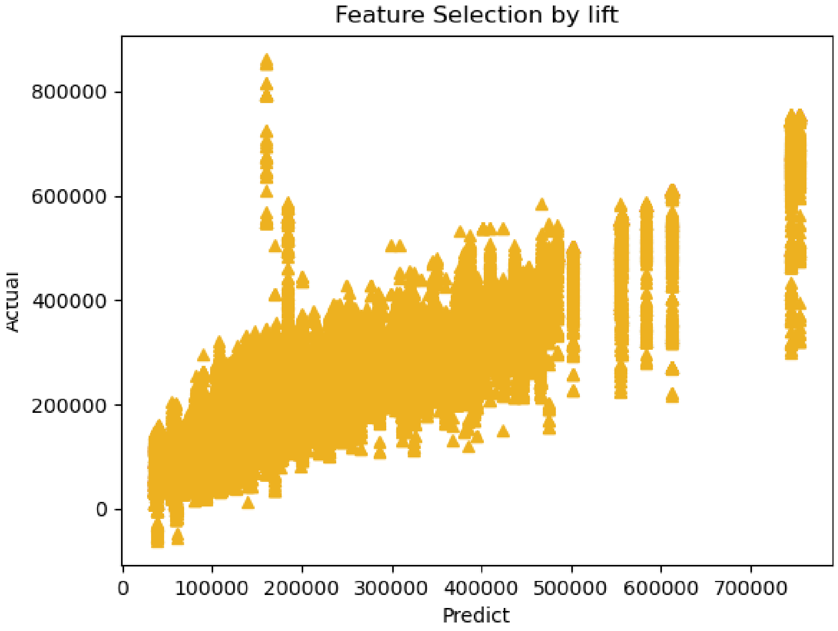

Figure 1: XGBoost Predictions Against Actual Prices Graph

Model	Feature Select	Support	Features	R-Square Score	MSE
Linear Regression	3	0.01	All Features	0.88	1783230
Linear Regression	7	0.01	Sort by Confidence	0.874	2004146
Linear Regression	10	0.1	Sort by Support	0.857	2271112
XGBoost	7	0.01	Sort by Confidence	0.962	593558.4
XGBoost	3	0.1	Sort by Lift	0.911	1414398
XGBoost	All	0.1	All Features	0.99	158669.7

Table 2: Sample Results on Diamonds Dataset

6.3 NBA Rookie

Testing the House Price dataset led to the results at table 3. For example, while training the Linear Regression model with top 10 features sorted by lift with support of 0.01, the best R-Square score of testing it was 0.1392 and MSE 0.204041875. You also can see the graph of XGBoost sorted by lift with a selection of top 10 features at Fig 2.

Model	Feature Select	Support	Features	R-Square Score	MSE
Linear Regression	All	0.01	All Features	0.203112461	0.188919
Linear Regression	10	0.01	Sort by Support	0.197598093	0.190224712
Linear Regression	10	0.01	Sort by Lift	0.139201388	0.204041875
XGBoost	All	0.1	All Features	0.959279309	0.009641
XGBoost	10	0.1	Sort by Support	0.950976478	0.011607904
XGBoost	10	0.1	Sort by Confidence	0.950005784	0.011841195
XGBoost	7	0.1	Sort by Lift	0.924530836	0.01787833
XGBoost	10	0.1	Sort by Lift	0.946882908	0.012583133

Table 3: Sample Results on NBA Rookie Dataset

6.4 Avocado Price

In the avocado pricing dataset, we've got a maximum R-Square score of 0.97 and average MSE 0.0028 with the XGBoost model trained by the data that includes all of the features and support of 0.01. All the K's led to this result but with different Mean square error. You can see a graph with the predictions for K = 3 in Figure 3. For K = 5 and sorting features by confidence gave R-Square score of 0.91. Testing the dataset with Linear Regression model gave a maximum R-Square score of only 0.63 with training it by all of the features or 0.49 with training it only by top 5 features selected by lift. The MSE co-responding to the best R-Square score is 0.052. If we raise the support to 0.1, The best R-Square score will be the same - 0.63 if we use the Linear Regression model or 0.88 for K = 7 and it's mean square error is 0.016.

you can see the best results at Table 4.

Model	Feature Select	Support	Features	R-Square Score	MSE
Linear Regression	5	0.01	All Features	0.63	0.052
Linear Regression	7	0.01	Sort by Confidence	0.61	0.055
XGBoost	3	0.01	All Features	0.97	0.002
XGBoost	7	0.01	Sort by Lift	0.95	0.006
Linear Regression	7	0.1	Sort by Support	0.59	0.057
XGBoost	0.1	3	All Features	0.97	0.002

Table 4: Best Results on Avocado Dataset.

7 Discussion and Future work

7.1 Hose Price

After calculating the results, we've found out that our approach is better when we use all of the features. AFRS with 10 features to select, support of 0.01, a XGBoost model and sorting by lift gives a 0.997 R-Square score and 1691821.64

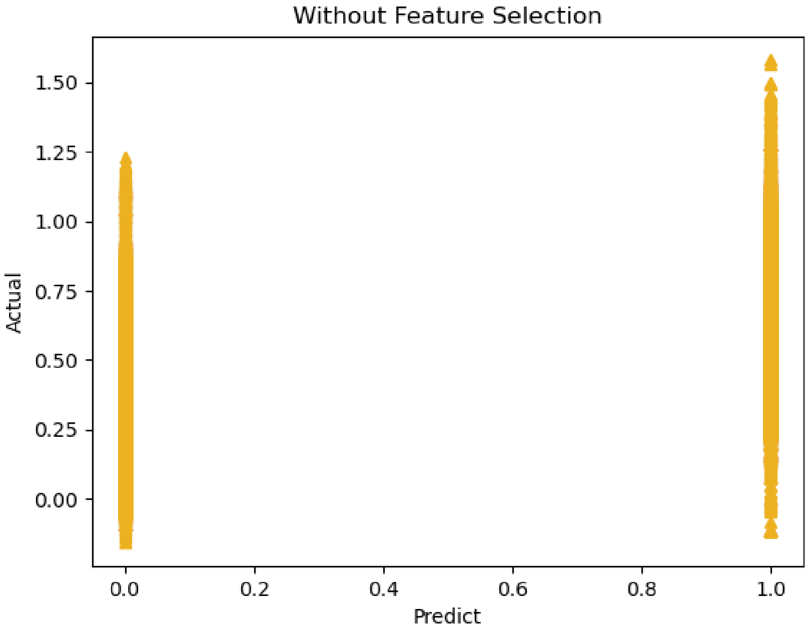

Figure 2: XGBoost Predictions Against Actual Prices Graph

MSE. Predicting with all of the features with the same model gives a 0.971 R-Square score and 17344051.21 MSE, like you can see at table 1. In addition, the same model sorted by support is also better than using all of the features.

7.2 Diamond Price

After calculating the results, we've concluded that training XGBoost model with all the features, along with support threshold of 0.01 leads us to the best prices predictions with a R-Square score of 0.99. The interesting this in the results of this experiment is that if we look at the same model and same threshold, but with training it by data that includes only the top 5 features by lift, we can see that the R-Square score is 0.959. So, maybe if we can take a little risk, we can train the model with less features of data (data with 5 features is better than 10 features) and by that the training process will be faster.

7.3 NBA Rookie

In this dataset we've found that our approach is not useful. The naive classification with all of the features using the XGBoost model gives a R-Square score of 0.959. In the other hand, after calculating the results of our tests, the ARFS

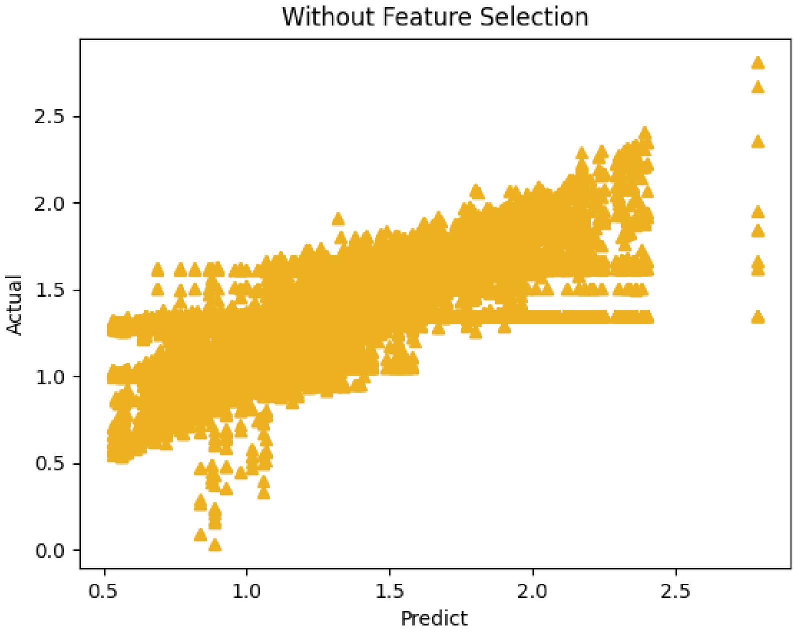

Figure 3: XGBoost Predictions Against Actual Prices Graph

leads to the highest R-Square score of 0.950 when sorting it by support. you can see at table 3.

7.4 Avocado Price

In the avocado pricing dataset, we've got a maximum R-Square error of 0.97 with the XGBoost model trained by the data that includes all of the features. But in the meaning of support threshold, we've seen that finding rules with threshold of 0.01 and threshold of 0.1 have led us to very similar results when we are training with all of the features. So, it seems that if we choose to train the machine with data that contains all of it's features, then using a higher support threshold (e.g. 0.1) can work good and give us rules faster.
On the other hand, if you don't want to use all of the features, but rather only a small amount of them, in this case working with 3 or 5 top K features will give us worse scores. In order to get scores that are as close to 0.97 as possible, using a support threshold of 0.01 is important but also the higher the K is, the better. For example, if we choose to train our machine only with the top 3 features (support 0.01), the score will be around 0.75, but when choosing k to be 10 brings us to the score of 0.96.

7.5 Conclusion and Future work

For conclusion our ARFS algorithm not always useful. We can see in classification problem our algorithm not help and the same results we get on other datasets. The most significant discovery was in House Price dataset, we think the reason is this dataset have 80 feature. It is not impossible that on datasets with lot of feature our algorithm can be useful. For future work we need to calibrate the number of feature we select and test more support threshold to find better resultd.

8 Code

We built a website with streamlit library. You can see our code project on Github. see also our google colab notebook.

References

[1] N. Petrick R.F. Wagner B. Sahiner, H.P. Chan and L. Hadjiiski. "feature selection and classifier performance in computer-aided diagnosis: The effect of finite sample size". *Medical Physics*, 27(7):1509–1522, 2000.

[2] K. Bailly and M. Milgram. Boosting feature selection for neural network based regression. *Neural Networks*, 22(5-6):748–756, 2009.

[3] C.-H. Wu C.-C. Lai and M.-C. Tsai. Feature selection using particle swarm optimization with application in spam filtering. *International Journal of Innovative Computing Information and Control*, 5(2):423–432, 2009.

[4] M. Dash and H. Liu. Technical report. *SIntell. Data Anal.*, pages 131–156, 1997.

[5] N. R. Draper and H. Smith. *Applied Regression Analysis*, volume 27(7). New York: Wiley, 2nd ed. edition, 1981.

[6] M. K. Jeong et al. E. Youn, L. Koenig. Support vector-based feature selection using fisher's linear discriminant and support vector machine. *Expert Systems with Applications*, 37(9):6158–6156, 2010.

[7] M. E. Elalami. A filter model for feature subset selection based on genetic algorithm. *Knowledge Based Systems*, 22(5):356–362, 2009.

[8] P. Langley et al. *Proceedings of the AAAI Fall Symposium on Relevance*. 1994.

[9] R. Kohavi G.H. John and K. Pfleger. *Machine Learning Proceedings*. Elsevier, 1994.

[10] S. Gunal and R. Edizkan. Subspace based feature selection for pattern recognition. *Information Sciences*, 178(19):3716–3726, 2008.

[11] Jaiwei Han and Micheline Kamber. *"Data Mining Concepts and Techniques"*. Morgan Kaufmann publishers, second edition edition.

[12] S. Hengpraprohm and P. Chongstitvatana. Feature selection by weighted-snr for cancer microarray data classification. *International Journal of Innovative Computing Information and Control*, 5(12(A)):4627–4635, 2009.

[13] N. Saenz-Lechon et al. J. I. Godino-Llorente, J. D. Arias-Londono. An improved method for voice pathology detection by means of a hmm-based feature space transformation. *Pattern Recognition*, 43(9):3100–3112, 2010.

[14] Jerzy S. J. Jelonek. "feature subset selection for classification of histological images. artificial intelligence in medicine". *Journal of Machine Learning Research*, 9:22–238, 1997.

[15] L.A. Rendell et al. K. Kira. Aaai. pages 129–134, 1992.

[16] M. Karabatak and M. C. Ince. A new feature selection method based on association rules for diagnosis of erythemato-squamous diseases. *Expert Systems with Applications*, 36(10):12500–12505, 2009.

[17] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[18] D. Koller and M. Sahami. Technical report. *Stanford InfoLab*, 1996.

[19] N. Ghasem-Aghaee M. H. Aghdam and M. E. Basiri. Application of ant colony optimization for feature selection in text categorization. *IEEE Congress on Evolutionary Computation*, pages 2867–2873, 2008.

[20] K.Z. Mao. "fast orthogonal forward selection algorithm for feature subset selection". *IEEE Transactions on Neural Networks*, 13(5):1218–1224, 2002.

[21] J. A. Gamez P. Bermejo and J. M. Puerta. A grasp algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. *Pattern Recognition Letters*, 32(5):701–711, 2011.

[22] J. Reunanen. "overfitting in making comparisons between variable selection methods". *Journal of Machine Learning Research*, 3 (7/8):1371–1382, 2003.

[23] C. C. Reyes-Aldasoro and A. Bhalerao. The bhattacharyya space for feature selection and its application to texture segmentation. *Pattern Recognition*, 39(5):812–826, 2006.

[24] A. Unler and A. Murat. A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research*, 206(3):528–539, 2010.