Advertisement images classification

Zijian Wang
zijian.wang@cranfield.ac.uk

Abstract

Classifying advertisement images on websites could filter unnecessary information for users. In the manuscript, three different machine learning algorithms are adopted to achieve the classification task, including decision tree, k-nearest neighbor and support vector machine. The initial data are shuffled and divided into training and test datasets. The training dataset is used to choose the most suitable parameters. And validation dataset is to evaluate the true performance of each model. The final experiment shows that SVM owns the highest accuracy with 97.03%, while its efficiency is not as good as others. Meanwhile, the recall of all models are under 90% due to the limited volume of positive samples. Finally, the required questions are answered specifically in the last section.

1 Introduction

Detecting advertisement images on web pages is useful to block unnecessary information for users.

1.1 Workflow

The workflow of classification tasks is shown in figure 1. First, raw data could be preprocessed, like filling in empty values and shuffling the data. Then feature engineering is to stress the important features manually. After that, the training data are used to search out the best parameters. Lastly, test data are inputted into the model and evaluates the performance.

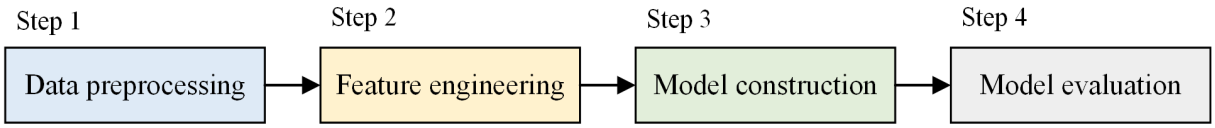


Figure 1: Work of classifying images

1.2 Evaluation criteria

Statistical criteria are adopted to measure a model's performance. For binary classification, confusion matrix in table 1 is very useful. In detail, false positive is to wrongly classify a positive sample; true positive is to correctly classify a positive sample; false negative is to wrongly classify a negative sample; true negative is to correctly classify a negative sample.

Accuracy is the proportion of all correctly classified samples to the total population at equation 1. Recall is the fraction of the total amount of relevant instances that are actually retrieved [1] at equation 2. Precision is the fraction of relevant instances among the retrieved instances [1]

Table 1: Confusion matrix.

	True conditions	
Predict conditions	True positive (TP)	False positive (FP)
	False negative (FN)	True negative (TN)

at equation 3. F1 considering both precision and recall could present its actual performance of classifying at equation 4.

$$Accuracy(ACC) = \frac{TP + TN}{Total \quad population} \tag{1}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{2}$$

$$Precision(P) = \frac{TP}{TP + FP} \tag{3}$$

$$F1 = \frac{2}{recall^{-1} + precision^{-1}} = \frac{2TP}{2TP + FN + FP} \tag{4}$$

2 Preprocessing and feature engineering

2.1 Data preprocessing

The raw file contains 2359 pieces of advertisement data. Each piece of data consists of 1559 dimensions. The first three dimensions contain float values varying from 1 to 600. The last dimension is strings, like 'ad.' and 'nonad.'. Other dimensions are binary values, as shown in table 2.

Table 2: Data sample

149	182	1.2214	0	0	0	.....	1	0	0	ad.
32	328	10.25	0	1	0	.....	0	0	0	ad.
20	134	6.7	1	0	1	.....	1	0	0	nonad.
38	88	2.3157	0	0	0	.....	0	0	0	nonad.

Firstly, we should check the completeness of data and fill in all empty values. Then, the last dimension is strings, which are incompatible with others. Hence, the string 'ad.' is replaced by number '1', presenting the advertisement. And the string 'nonad.' is replaced by number '0'. Besides, the data is sorted by advertisement, which are not suitable for training and test. A shuffle function is used to random sort the data.

After that, 70% data are divided into a training dataset. The other 30% data belong to a validation dataset. The training data are used to train models and choose the best parameters. 10 fold cross validation splits the training data into 10 subsets, and each subset could be used to test the performance. Adopting 10 fold crossing validation is useful to avoid overfitting. Meanwhile, after building the model, the validation dataset which is not used in training could measure the true performance of the model, as shown in figure 2.

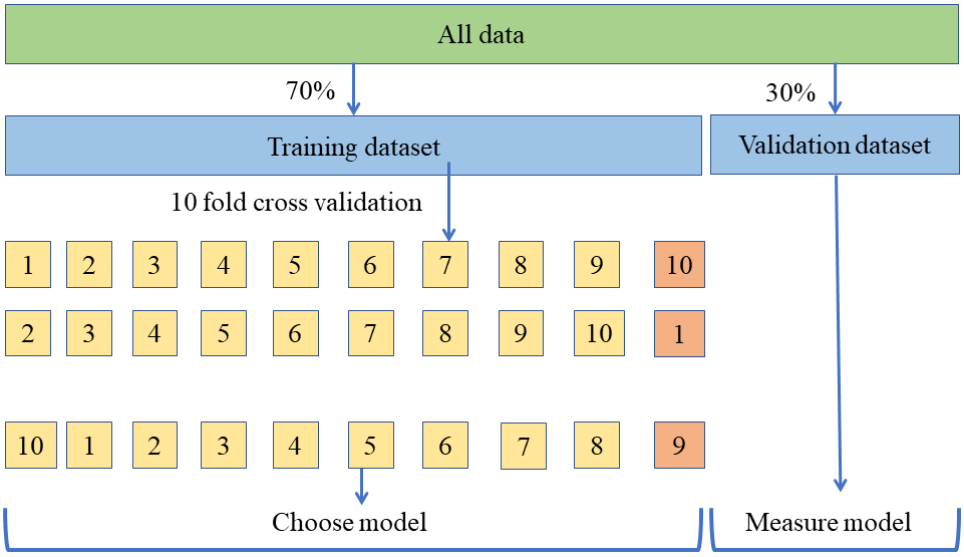


Figure 2: Construct datasets.

2.2 Feature engineering

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques [2]. These features can be used to improve the performance of machine learning algorithms [3]. In this task, the first three dimensions are not binary values. Their ranges are very large. We can use scale techniques to make sure the same range of each dimension's values.

3 Model and experiment

3.1 Decision tree

A decision tree is a basic machine learning algorithm for classification and regression. It was firstly presented by Quinlan, ID3[4] and C4.5 algorithm[5]. Decision trees have some advantages. It is easy to understand and the whole process could be visualized. The algorithm needs very little preparation and is easy to handle. Meanwhile, it is a white box containing all possible hypotheses. However, decision trees are easy to overfit, which are only good at dealing with training data. And it might be unstable due to the small variations in the data.

3.1.1 Construct a decision tree

The key idea is to choose features according to the information gain (in section 6.2), and construct a tree recursively. Firstly, the information gain of all possible features is calculated out. The one which could decrease the impurity of the whole system mostly is chosen as the node. Then, the same method is applied to select other nodes until there are no features.

Depth, the levels of a tree, could present its complexity. If we construct a decision tree until there is no feature, it might be overfitting. For avoiding this, we monitor the process of modeling as shown in figure 3. In this case, with the increase of depth, the accuracy of the training dataset always increases. However, the trend of the test dataset rises dramatically at first and then starts to drop at the depth of 13. Hence, the tree is started to be overfitting after the depth of 13. And we should stop building immediately.

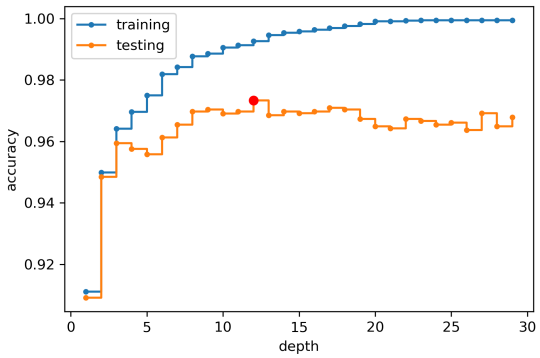


Figure 3: Accuracy with depth increase.

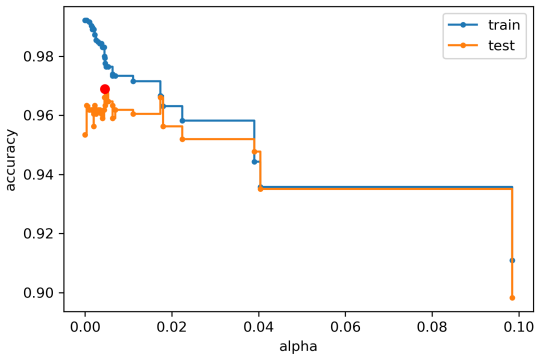


Figure 4: Accuracy with alpha increase.

3.1.2 Prune a decision tree

The previous process is the basic ID3 algorithm. After building a tree, there might be extra nodes. For furthermore avoiding overfitting and improving its robustness, pruning a tree is necessary. Alpha is another criterion of the complexity of decision trees. The small alpha presents a high-complexity model.

When the alpha begins to increase, nodes of the model are pruned. The accuracy of the training dataset keeps a decreasing trend. But the accuracy of the test dataset increases at the beginning and then drops dramatically to a low level, as shown in figure 4. When the tree is pruned too much, it loses the function to classify advertisement images. According to the performance of the test dataset, 0.01 is chosen as the suitable alpha to prune the tree.

3.2 KNN

K-nearest neighbor (KNN) is a supervised learning method, which is usually used in classification. For new input data, the model finds out the nearest samples in training data and classifies the input as one class according to the major samples. There are three important factors for building KNN models.

Firstly, distance measurement. Different methods could calculate out varied distances between two samples. If $x_i = (x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, ..., x_i^{(n)})$, $x_j = (x_j^{(1)}, x_j^{(2)}, x_j^{(3)}, ..., x_j^{(n)})$, then the distance called L_p is defined by equation (5):

$$L_p(x_i, x_j) = \left[\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p \right]^{\frac{1}{p}} \tag{5}$$

If $p = 2$, $L_2(x_i, x_j)$ is called Euclidean distance at equation (6):

$$L_2(x_i, x_j) = \left[\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^2 \right]^{\frac{1}{2}} \tag{6}$$

If $p = 1$, $L_1(x_i, x_j)$ is termed as Manhattan distance at equation (7):

$$L_1(x_i, x_j) = \left[\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}| \right] \tag{7}$$

Another important factor is the classification rule. And the majority voting rule is the most common one. It means that the class of an input is decided by the class of majority nearest samples, as shown in equation 8:

$$y = \arg\max_{c_j} \sum_{x_j \in N_k(x)} I(y_i = c_j), \quad i = 1, 2, ..., K; j = 1, 2, ..., N \tag{8}$$

Where I is indicator function. When $y_i = c_j$, I equals to 1.

The last factor is the value K, presenting the number of nearest neighbors. How many training samples should we choose when classifying a new input? If K is very small, the approximation error could be very small too. But, the model can only predict similar input data and lacks robustness. On the other hand, the K is too large, where the approximation error decreases. However, it is easy to predict input wrongly.

In this experiment, we choose the Euclidean distance as the measurement. Meanwhile, a range of K is set to search out the best one. For avoiding overfitting, cross validation is applied during parameter tuning processing.

3.3 SVM

Support vector machine is a supervised learning model for classifying two types. Its main idea is to find out a separator, which would be a line or a plane to separate two different types of objects. Meanwhile, it also needs to find out the max-margin, which could decrease the wrong prediction and improve its robustness, as shown in figure 5. These points on the margin, the two blue points and the green one in figure 5, are called support vectors.

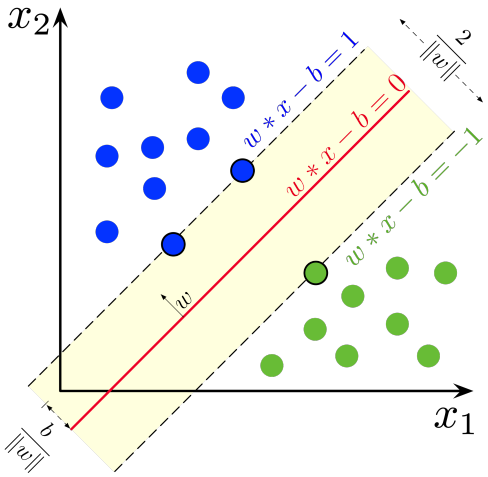
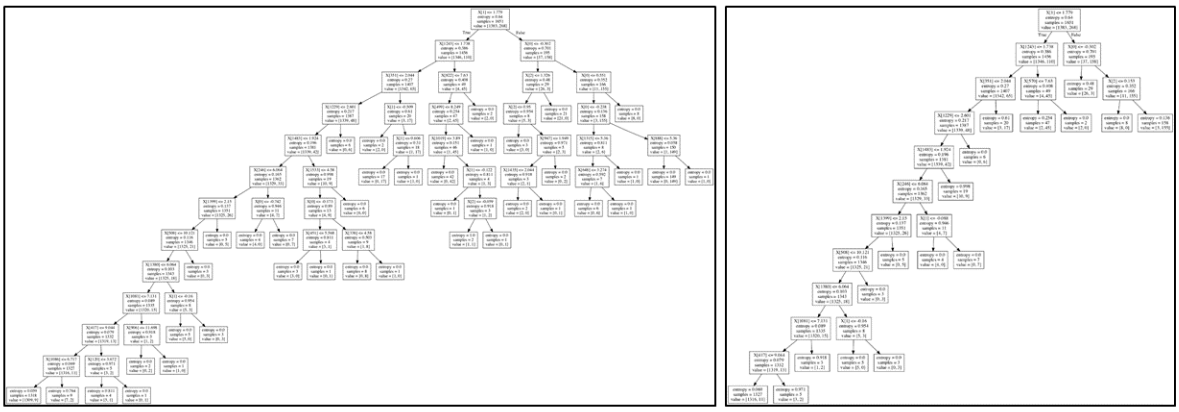


Figure 5: Support vector machine [6].

4 Discussion

4.1 Prune a decision tree

Decision trees have been presented in figure 6. The left one is the raw decision tree created by ID3 algorithm, where the depth is 13 and creates node until no features. The right one is pruned lightly according to C4.5 algorithm. Many unnecessary nodes in the decision tree have been erased. The whole tree becomes simple than before. Through pruning, the robustness has been improved and avoid overfitting.



(a) Before pruning (b) After pruning

Figure 6: Visualization of decision trees.

Meanwhile, figure 7 shows the performance of two decision trees. The recall of pruned decision tree increase by 2%. The accuracy and F1 also improve lightly. However, for precision, the raw decision tree is a little higher than the pruned one. In general, the pruned decision tree owns a better performance in robustness and prediction accuracy.

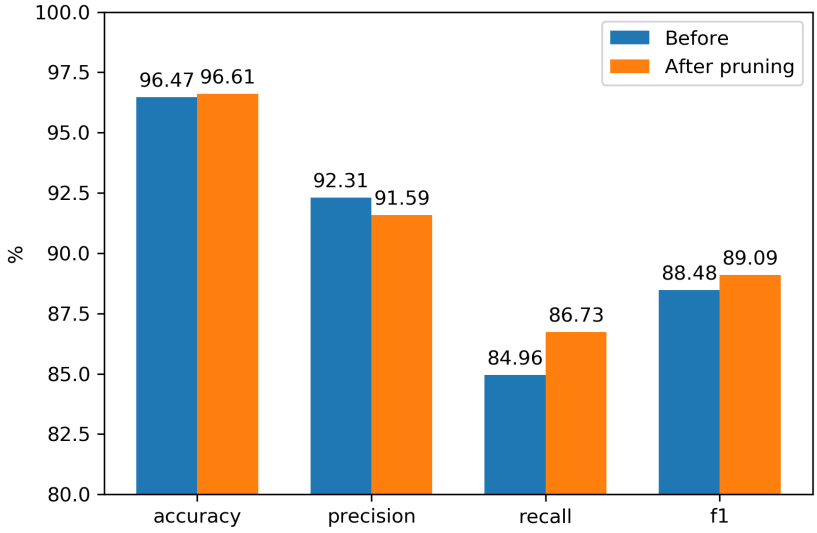


Figure 7: Performance compare of decision trees.

4.2 Performance of three methods

Accuracy, precision, recall, and F1 are adopted to evaluate the performance of models, as shown in figure 8. In general, SVM owns the best performance, while KNN is the worst one. However, the recall of all three models is not as good as other criteria. It is more possible to mispredict the advertisement images as non-advertisement ones, or called false positives. There are 381 pieces of advertisement image data which account for only 16% of the entire data. The limited positive data constricts the performance of models, which are poor at predicting positive ones. For further improvement, we could add more advertisement image data. Then, the recall and F1 could be improved too.

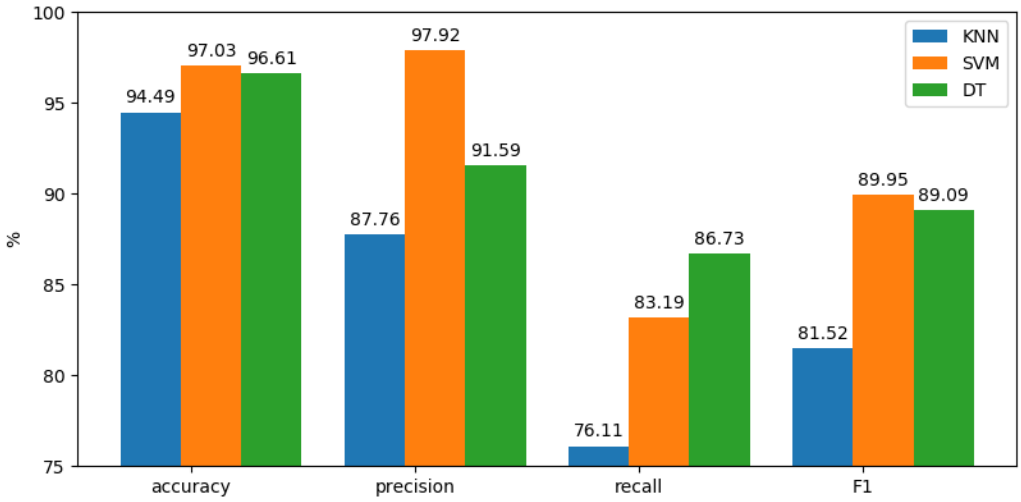


Figure 8: Performance compare of three methods.

4.3 Efficiency of SVM

Although SVM has the best performance, it costs a long time to train the model. In the experiment, SVM spends almost 30 minutes to find out the best parameters, which are dozens of times longer than the other two algorithms (normal laptop instead of a high-performance computer). The key work of building an SVM is to search out the hyperplane to divide different classes. In this case, each piece of data has nearly 1500 dimensions of values. Hence, it costs a long time to find suitable hyperplane with the considers of 1500 dimensions.

4.4 How to choose parameters

The performance of a model is affected by parameters. Selecting suitable parameters are other tasks after deciding models. Two different approaches to selecting parameters are used in this experiment. The first one is the grid search. The basic idea is to try each combination of possible parameters and export the best performance one automatically. In this task, KNN is constructed by this approach.

Another one is validation curves. Only one parameter is changed every time and the performance curves of training and test data are outputted, like figure 3 and figure 4. It needs humans to choose the parameters according to the curves. The SVM and DT are built by validation curves.

The advantage of the grid search is to select parameters automatically. For avoiding overfitting, cross validation could be adopted during the grid search. But due to the exhausting search, it costs much computational source. Moreover, the merit of validation curves is to visualize the performance and help researchers select parameters. It also occupies less computational sources because it changes one parameter once and saves much unnecessary computation.

5 Conclusion

Three machine learning methods, DT, KNN, and SVM, are utilized to classify advertisement images. The main conclusions are as following:

- Constructing decision trees include two processes: building a whole tree firstly and pruning unless nodes. Through pruning, its robustness increases greatly and the accuracy improves too.

- SVM owns the best performance compared with the other two methods. Especially, the accuracy of SVM is 97.03%. But training SVM costs the longest time because it's difficult to search out the hyperplane based on thousands of dimensions.
- The recall of all models are not as good as other criteria, which means that they could easily make FPs. For improving the performance, more positive samples could be added in.

6 Required questions

6.1 Create data sets

Data in machine learning usually are divided into two categories, training data, and test data. Training data are used to train the system. And test data are used to test the performance of the system. Usually, test data are not used during training.

Two-fold cross-validation is a simple way to evaluate machine learning models. The data could be split into two sets S1 and S2 randomly. A common way is to split 70% data into a training data set and the left 30% data could be as a test data set. S1 is used to choose models and adjust parameters. And the models are measured by S2.

Another approach is the K-fold cross validation. First, data could be split into k subsets randomly. Then (n-1) subsets are taken out to train the model and the last subset is utilized to evaluate the model. After that, the mean error of all possible k tests is reported for choosing the best performance one. Empirically, k = 10 is often adopted in many cases.

6.2 Entropy and information gain

A decision tree is built top-down from a root node and searches through the space of possible branches. ID3, the core decision tree algorithm, adopts $entropy$ and $informationgain$ to construct the model.

Entropy is a measure of impurity, as shown in equation 9. Entropy is typically measured in bit or nat. According to the definition, entropy is only related to the distribution of S instead of the value of S. The bigger entropy presents the larger impurity of the element.

$$E(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{9}$$

where, S is a sample of training examples; p_i is the probability of class i in s.

Information gain is the expected reduction in entropy due to splitting on attribute A. In other words, it presents a reduction in impurity in the data, as shown in equation 10 and 11.

$$g(D, A) = E(D) - E(D|A) \tag{10}$$

$$E(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} E(D_i) \tag{11}$$

For a given data D and an attribute A, entropy $E(D)$ means the impurity of data D. And $E(D|A)$ presents the impurity of data D under the splitting on attribute A. Hence, when reducing, $g(D, A)$ is the reducing impurity based on attribute A. Different distributions have different information gain. The large information gain usually owns a more powerful classification.

6.3 Inductive bias

Inductive bias is the set of assumptions that the learner uses to predict outputs given inputs that it has not encountered. For ID3 decision trees, the model prefers short-route trees. Meanwhile, the high information gain attributes could be toper than others. Besides, there is a set of all possible hypotheses, instead of a restriction of hypothesis space.

6.4 Overfitting

Overfitting is when the model performs very well on the training data but performs very badly on test data. Usually, when the training error could decrease and the test error starts to increase, there is overfitting.

There are many ways to avoid overfitting. One way is to increase the dataset. More data are used to train the model, and the model is hard to overfit all data. We could also stop training the model when starting overfitting and simplify the model. Optimum model complexity could be useful to judge a model's status, as shown in figure 9. When the model is more complex, the variance becomes large and the bias is small. And the total error drops slowly then increase rises with the increase of model complexity, like a U shape. Ideally, we could like to pick a perfect model that has been trained very well but not overfitting. This could be the bottom of total error, which owns low variance and bias.

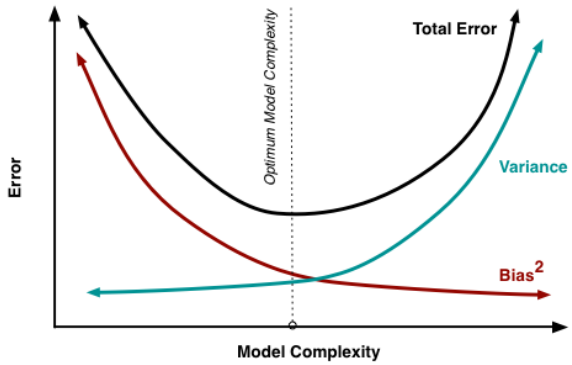


Figure 9: Optimum model complexity.

For decision trees, we could stop growing trees when data split cannot have a statistically significant performance. And we could prune unnecessary nodes after building trees.

6.5 Hypothesis

A hypothesis is the current function approximation where we are approximating an unknown target function that can best map inputs to outputs on all possible observations from the problem domain.

Hypothesis space is the set of all possible hypotheses from which the target function is selected through the machine learning process.

Meanwhile, learning is the process of searching through the hypothesis space for a single, simplest hypothesis that is consistent with the training data.

References

[1] Wikipedia contributors, F1 score — Wikipedia, the free encyclopedia, 2020. URL: `https://en.wikipedia.org/w/index.php?title=F1_score&oldid=947582090`, [Online; accessed 29-March-2020].

[2] C. R. Turner, A. Fuggetta, L. Lavazza, A. L. Wolf, A conceptual basis for feature engineering, Journal of Systems and Software 49 (1999) 3–15.

[3] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, 2019.

[4] J. R. Quinlan, Induction of decision trees, Machine learning 1 (1986) 81–106.

[5] J. R. Quinlan, C4. 5: programs for machine learning, Elsevier, 2014.

[6] Wikipedia contributors, Support-vector machine — Wikipedia, the free encyclopedia, 2020. URL: https://en.wikipedia.org/w/index.php?title=Support-vector_machine&oldid=943556714, [Online; accessed 28-March-2020].