National University of Ireland, Galway

Iva Simon Bubalo
Student ID:
202***

Programme:
2021-1MAO2: MAO2
Master of Science in Computer Science - Artificial Intelligence

Module:
Machine Learning

Classification of beer styles (stout, lager, or ale) using supervised learning methods

Introduction

The objective of this report is to select and apply two different classification algorithms to a dataset and compare the results. Two classification algorithms to be described in this report are k-Nearest Neighbour (kNN) and Support Vector Machine (SVM). Both kNN and SVM belong to the family of supervised learning algorithms.

Dataset description

The dataset used in this report is provided by NUIG and is pre-split into training and testing subsets. Training dataset contains 124 rows and 10 columns, while testing contains 30 rows and 10 columns. Features listed in the dataset are: *calorific value, nitrogen, turbidity, style, alcohol, sugars, bitterness, beer id, colour,* and *degree of fermentation*. The target variable for classification will be the *style* column.

Machine learning package used

On the open-source market there are several machine learning packages used for different types of machine learning tasks. The most notable ones are TensorFlow, Scikit-learn, Keras, PyTorch, and Spark MLib. The majority of these packages also contain deep learning algorithms. I chose Scikit-learn for its simplicity, wide community support, focus on machine learning exclusively, as other packages seemed to be more appropriate for more complex tasks.

Data preparation

Pre-processing phase consisted of the following steps:

- determining there was no missing values to impact the models,
- separating the target variable *style* from the training and testing datasets into *y_train* and *y_test* arrays,
- categorical label encoding of the target variable *style* models, which converts three unique values *ale*, *lager*, and *stout* to numerical values 0, 1, and 2,
- at a later stage, normalizing the scale for all features in training and testing datasets.

Limitations of label encoding and scaling will be discussed in the *Results discussion* section.

Supervised learning algorithms applied

Supervised machine learning techniques generate a model of the relationship between a set of descriptive variables and a target variable based

on a set of historical examples, or instances.[1] Classification algorithms that belong to the family of supervised learning algorithms, are used when the output variable is restricted to a limited set of values, for example we may want to classify weather as *sunny*, *cloudy*, or *rainy*, so a task of predicting whether an object belongs to a category is called classification task.[2]

k-Nearest Neighbor (kNN)

Neighbors-based classification methods are known as non-generalizing supervised machine learning methods. They simply store, or "remember", all of its training data without attempting to construct a general internal model. Neighbors-based classification algorithms determine a predefined number of training samples closest in distance to the new data point, and predict the label from these. The number of samples can be a user-defined constant (*k-nearest neighbor*), or vary based on the local density of points (*radius-based neighbor* learning). The distance can, in general, be any metric measure, where standard Euclidean distance is the most common choice. [3]

k in kNN is a parameter that refers to the number of nearest neighbours to include in the majority vote process. Classification is computed from majority vote of the nearest neighbors of each data point.[4] A new data point will be assigned a class according to the number of its closest representatives (the nearest neighbors). For example, for *k*=3 the labels of the three closest classes are checked and the most commonly occurring label will be assigned to the new data point.

Support Vector Machine (SVM)

Support Vector Machines (SVMs) are a set of supervised learning methods. In principle, SVMs construct a hyperplane or set of hyperplanes in a high-dimensional space that separate data points belonging to different classes. SVMs can perform both linear and non-linear classification. The best fitting hyperplane is considered the one that represents the largest separation of data points, or largest margin between two or more classes. [5]

Results Discussion

As mentioned earlier in the pre-processing step, label encoding converts categorical text data into model-understandable numerical data, for example 0, 1, 2, etc with no relationships between them. There may be some limitations and new problems that arise with label encoding, such as the model assuming that there is an order or hierarchy between those numbers.[6]

A potential alternative approach would be to use *One Hot Encoder* from Scikit-learn that splits the target variable into multiple columns and assigns 1s and 0s depending on which column has that value. While one hot encoding solves the problem of incorrect relationships within a variable, it can also significantly increase the dimensionality of the data. With the increase in dimensionality, new problems, such as **the curse of dimensionality**[7], may arise in machine learning, which is beyond the current scope of this report.

k-Nearest Neighbor (kNN) results

First attempt in model building was done prior to normalizing the scale of training variables, and it resulted in low model performance for kNN (k = 3) and accuracy of 46%. In the following step I used *scale* function from Scikit-learn to bring all features to the same scale. The newly trained model on scaled data achieved the accuracy of 97% for training dataset, and 93% for testing.

---

[1] Kelleher, J.D., Mac Namee, B., D'Arcy, A. (2015) *Fundamentals of Machine Learning for Predictive Data Analytics*, MIT Press Cambridge

[2] Russell, S., Norvig, P. (2010) *Artificial Intelligence, A moder Approach*, (Third ed.) Prentice Hall, Pearson Education, Inc.,New Jersey 07458.

[3] Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R. (2005) '*Neighbourhood Components Analysis'*, *Advances in Neural Information Processing Systems*, Vol. 17, pp. 513-520.

[4] Ibid.

[5] Hastie, T., Tibshirani, R., Friedman, J. (2008). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction* (Second ed.). New York: Springer.

[6] scikit-learn.org "Preprocessing data". Online. Available at: https://scikit-learn.org/stable/modules/preprocessing.html#encoding-categorical-features (Accessed 1 Nov 2020)

[7] Bellman, R. E. (2003). *Dynamic Programming.* Courier Dover Publications.

```
k-NN score for test set: 0.933333
k-NN score for training set: 0.975806
              precision    recall  f1-score   support

           0       0.91      1.00      0.95        10
           1       0.90      0.90      0.90        10
           2       1.00      0.90      0.95        10

    accuracy                           0.93        30
   macro avg       0.94      0.93      0.93        30
weighted avg       0.94      0.93      0.93        30
```

Figure 1. Classification report for kNN (k=3)

Since *k* value is often user-defined, finding the optimal value of *k* can sometimes be challenging. It is a trade-off between noise in data having a higher influence on the result for small *k* values and large *k* values overfitting the data and becoming computationally expensive.

Cross-validation allows us to find a point in which overfitting begins to happen. Cross-validation technique divides data into k number of partitions of equal size, and iteratively performs the analysis on the training set and the validation on the testing set. Finally, the *k* sets of performance measures are aggregated to give one overall set of performance measures.[8]

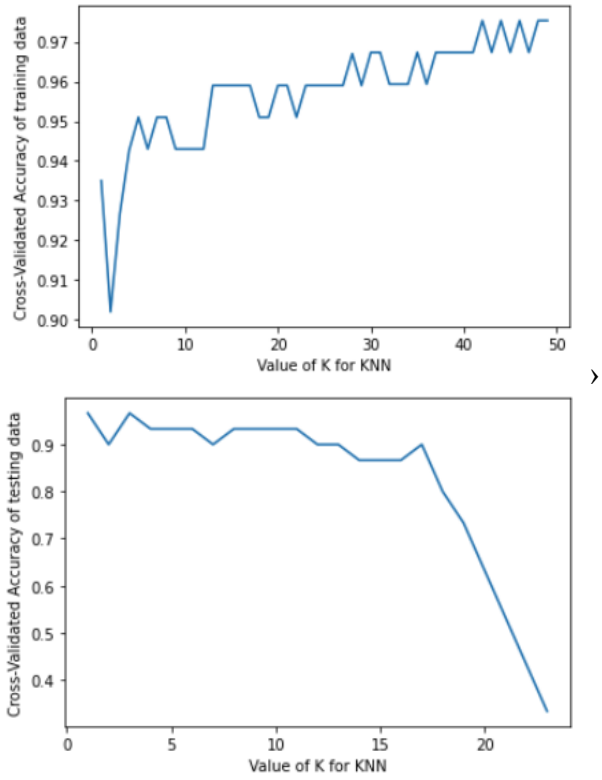

Figure 2. Cross-validation of different values of *k*

The figure above shows cross-validated accuracy for a range of different k values in training and testing datasets. We can clearly see that the decrease in performance of the testing data after *k* = 15, therefore in the training data for *k* over 15 the model starts to overfit.

Support Vector Machine (SVM) results

SVMs can use different kernels to separate classes; linear and non-linear. Typically, at the beginning of the model build we may not know which kernel will perform better so I followed the Occam's Razor rule of trying the simplest approach first, with linear kernel, then ran the same model with RBF kernel.

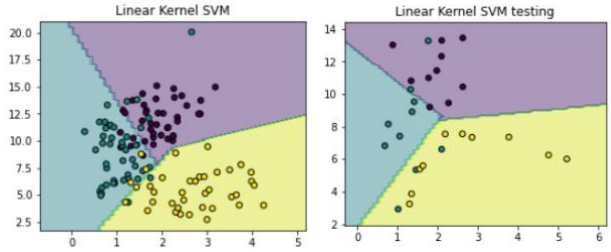

Figure 4. SVM with linear kernel and non-normalized training and testing data with the first 2 features: *turbidity* and *sugars*

I proceeded by experimenting with both normalized and non-normalized data. Linear kernel does not seem to be affected by the normalization, achieving the accuracy of 93%, while RBF kernel seems to be highly dependable on normalization. SVM with non-normalized data achieved only 27% accuracy, while declaring the same model with normalized data resulted in 97% accuracy.

[8] Kelleher, J.D., Mac Namee, B., D'Arcy, A. (2015) *Fundamentals of Machine Learning for Predictive Data Analytics*, MIT Press Cambridge

	precision	recall	f1-score	support
0	0.38	0.30	0.33	10
1	0.29	0.50	0.37	10
2	0.00	0.00	0.00	10
accuracy			0.27	30
macro avg	0.22	0.27	0.23	30
weighted avg	0.22	0.27	0.23	30

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	0.91	1.00	0.95	10
2	1.00	0.90	0.95	10
accuracy			0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

Figure 5. SVM with RBF kernel and non-normalized data (top), and normalized data (bottom)

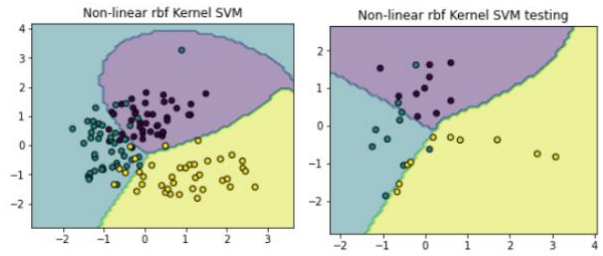

Figure 6. SVM with RBF kernel and normalized training and testing data with the first 2 features: *turbidity* and *sugars*

Both kNN and SVM models have reached the same maximum accuracy of 97%, however, accuracy is not the only performance metric in machine learning. SVM is slightly more precise in predicting ale and lager styles, and has a 10 percentage points better recall for lager style.

References

Bellman, R. E. (2003). *Dynamic Programming*. Courier Dover Publications.

Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R. (2005) '*Neighbourhood Components Analysis'*, *Advances in Neural Information Processing Systems*, Vol. 17, pp. 513-520.

Hastie, T., Tibshirani, R., Friedman, J. (2008). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction* (Second ed.). New York: Springer.

Kelleher, J.D., Mac Namee, B., D'Arcy, A. (2015) *Fundamentals of Machine Learning for Predictive Data Analytics*, MIT Press Cambridge

Russell, S., Norvig, P. (2010) *Artificial Intelligence, A moder Approach*, (Third ed.) Prentice Hall, Pearson Education, Inc.,New Jersey 07458.

scikit-learn.org "Supervised Learning". Online. Available at: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning (Accessed 1 Nov 2020).

scikit-learn.org "Preprocessing data". Online. Available at: https://scikit-learn.org/stable/modules/preprocessing.html#encoding-categorical-features (Accessed 1 Nov 2020).