Machine Learning Report

Cristian Sirbu
Cand No: 230982

Table of Contents

OVERVIEW

Classification is the process of categorising elements based on their similarities. The idea is to make each categorization cell as comparable as feasible (to minimise within group variance). This might imply maximising inter-group variation by increasing the distance between cells. A class is a particular cell or category inside a broader categorization. As in "The classification process generated an exceptional classification," the term "classification" refers to both the procedure and the product of the process. A proper categorization must be both mutually exclusive and exhaustive. In other words, the categorization must assign a spot (but only one) to everyone in the sample. There are two types of supervised classification problems: binary classification (with just one target variable) and multiple classifications (with several target variables).

For the problem presented for this report I will use a binary classifier. Binary classification is dichotomization in action. In many real binary classification situations, the two groups are not symmetric, and rather than total accuracy, the relative proportion of different sorts of mistakes is of relevance.

INTRODUCTION

For this paper I have been provided with two training data sets and one testing set. Training sets contains features which came from photos after applying a convolutional neural network to extract them. The second data set has some missing data which is indicated as Nan (not a number). My mission is to classify the sample data into memorable and not memorable. Every picture depends on the CNN and Gist future if its memorable or not.

In this case I chose to use SVM to perform this classification. SVM, or Support Vector Machine, is a linear model that may be used to solve classification and regression issues. It can handle linear and nonlinear problems and is useful for a wide range of practical applications. The concept of SVM is straightforward: The method draws a line or a hyperplane to divide the data into classes.

In order to use this classifier firstly we need to scale the data and for the second case where we have Not a Number cells, we need to replace them with a number of type float64. To be more accurate about that I decided to add the mean value of the features. After normalisation we can easily apply a classifier from sklearn and see how our training data performs.

How They Work

As I said earlier, I decided to use a Support Vector Machine to do this linear classification but to get a better accuracy I need to see the weights of importance of each column for my model. Firstly, my approach was to use SelectKBest from sklearn.future_selection to do the selection of the features but because of lack of features I was losing a good amount of accuracy and I decided to use PCA.

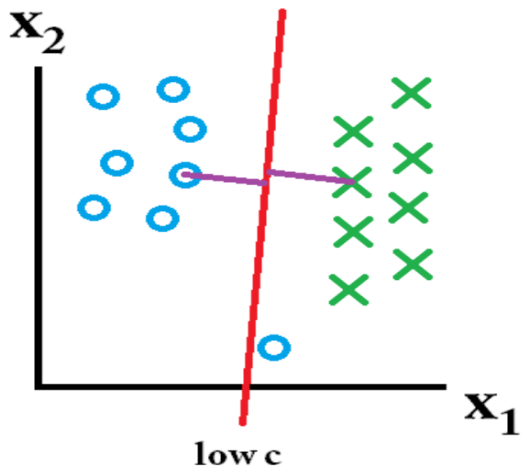
Principal Component Analysis is a linear dimensionality reduction approach that may be used to extract information from a high-dimensional space by projecting it onto a lower-dimensional sub-space. In the end I decided to reduce my futures to 50 which gave me an accuracy of 70 precents.

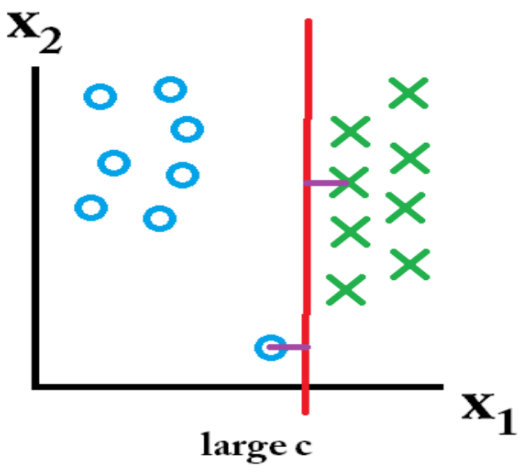
For each training set I delimit my CNN and Gist data. To normalise the data, I used a scaler. Normalization is important when your data has variable scales, and my model does not make assumptions about the distribution of your data.

To see how the default SVM classifier from sklearn works I trained and tested on my first training set which has all the outputs from CNN and Gist.

Model

A SVM seeks two things: a hyperplane with the greatest minimum margin and a hyperplane that accurately separates as many instances as feasible. The issue is that you won't always be able to acquire both. The C parameter influences your level of desire for the latter. I've sketched a simple example to demonstrate this. A low C on the left provides you a rather substantial minimum margin (purple). However, we must disregard the blue circle outlier that we failed to correctly categorise. You have a high C on the right. You will no longer ignore the outlier, resulting in a significantly smaller margin.





The C parameter indicates to the SVM optimizer how much you wish to prevent misclassifying each training example. For large values of C, the optimization will select a smaller-margin hyperplane if it performs a better job of accurately classifying all the training points. A very small value of C, on the other hand, will encourage the optimizer to seek a larger margin separating hyperplane, even if that hyperplane misclassifies more points. Misclassified cases should be expected for very small values of C, even if your training data is linearly separable.

To show how the number of C and Gamma influence my training data I performed some classifications. The result of those classification you can see below this paragraph.

C[1, 10, 100]:

```
[array([0.70892857, 0.74821429, 0.71428571, 0.74821429, 0.76428571])
```

G[0.001, 0.01, 0.1]:

```
array([0.70892857, 0.74821429, 0.71428571, 0.74821429, 0.76428571]),
```

As we can notice our data set is well delimited and changing the value of C and gamma does not influence our accuracy overall. After checking the accuracy for different values of C and Gamma and observing the outcome I decided to use C=10 and Gamma = 0.001.

As a technique for estimating the performance of my model I used cross validation. Cross-validation is a resampling technique used to assess machine learning models on a small sample of data. The process contains a single parameter called k that specifies the number of groups into which a given data sample should be divided.

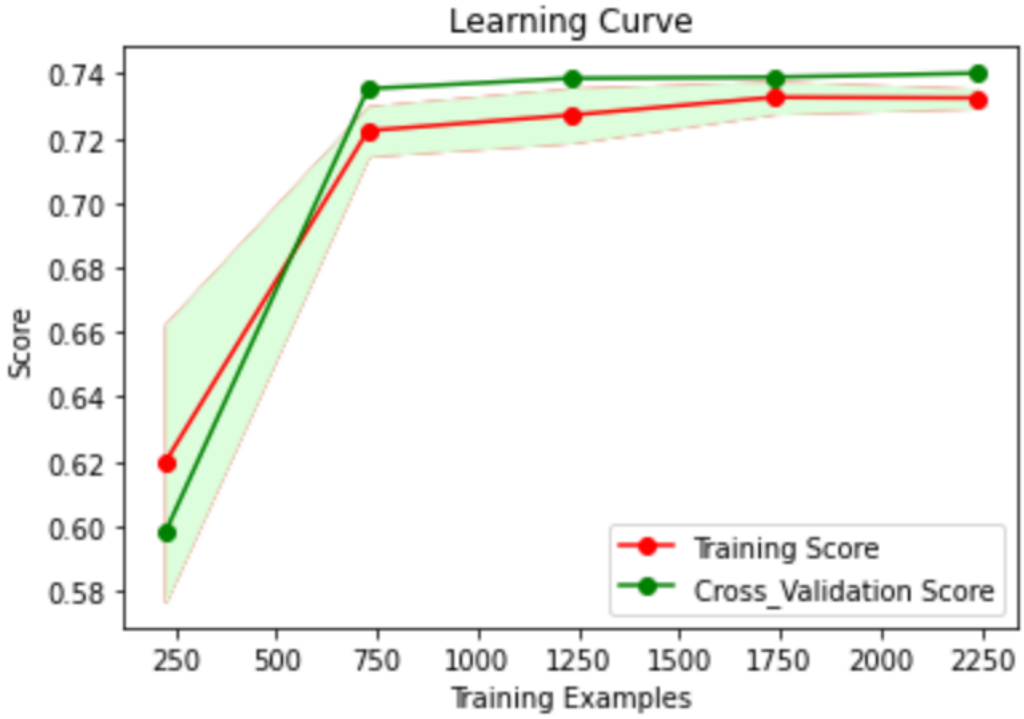
```
### SVM
k = svm.SVC(kernel="linear", C=10, gamma=0.001)
k.fit(X,y)
predictionsK = k.predict(CNN_test_label)

### SVM accuracy
scores = cross_val_score(k, X, y, scoring="accuracy")
np.mean(scores)
```

```
0.7375
```

After using this technique my model scored 73 precent on the test labels.

CONCLUSION

It is easy to preform classification since sklearn has all the classifiers premade, but we need to pay attention to the data and the format of it. For this paper I have given more than 4000 features. Using them all will make our model to overfit, and vice versa. So, I decided to split the features using Principal Component Analysis because this method keeps the most important features. After using SelectKBest I noticed that features which comes from Gist are not important in classifying if the image is memorable or not. Also, I decided to fill not number values from the second data set with the mean value.



For scoring I decided to use Cross Validation function provided from sklearn, and as we can see in the image above it increases the accuracy with the number of samples.

Also, a big lesson that I took from this report is how important is the Normalisation of the data before sending it to a classifier. Choosing the model, it has a big impact on accuracy but because we had to do binary classification, I checked my data with Logistic Regression and Random Forest to see if one of them outperforms SVM.

To score the accuracy of these type of models I used Cross Validation. The difference between cross_val_score and cross validation as model selection method. cross_val_score as name suggests, works only on scores. Confusion matrix is not a score, it is a kind of summary of what happened during evaluation. And maybe if I used confusion matrix, I would notice some improvements to do to score better.

REFERENCES

- *Cross-validation - Wikipedia* (no date) *Cross-validation - Wikipedia*. en.wikipedia.org. Available at: https://en.wikipedia.org/wiki/Cross-validation (Accessed: May 21, 2022).
- *machine learning - What is the influence of C in SVMs with linear kernel? - Cross Validated* (2012) *Cross Validated*. stats.stackexchange.com. Available at: https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel (Accessed: May 21, 2022).
- Brownlee, J. (2016) *Logistic Regression for Machine Learning - Machine Learning Mastery*, *Machine Learning Mastery*. machinelearningmastery.com. Available at: https://machinelearningmastery.com/logistic-regression-for-machine-learning/ (Accessed: May 21, 2022).
- *SVM* (no date) *https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72*. Available at: https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72 (Accessed: May 21, 2022).