Student Number: 184521 & Kaggle Team Name: Cand184521

1. Introduction

In this report I aim to look at binary classification and see how preprocessing and different classifier models affect the recall, precision and accuracy of classifying images. The two classifiers that have been chosen for this are Multi-Layer Perception (MLP) and Random Forests (RF) due to some of their opposing pros and cons for the task of looking at different photos of London and classifying them as memorable or not memorable based on given training data from Brighton with the same labels.

2. Methodology

2. 1. MLP Classifier

Initially an MLP classifier must be built to see how it operates without any preprocessing or additional data being added. This shall be done with the sklearn library which contains an MLPClassifier and will therefore need to be imported as well as other functions necessary to preprocess and sort the data.[1][3] There are also imports that give data to show the current performance of the classifier which will allow us to work out how the data maybe preprocessed to increase performance. Therefore, classes such as the standard scaler will have to be imported so that the data can be standardized and allow the classifier to perform better. Another addition that could help improve the MLP is more data which can be retrieved from the additional training file. This file will have to be pre-processed as we will only need enough data to balance out any difference between non-memorable and memorable examples. This means that the data will need to be split and reduced to fit this standard, also a simple imputer is needed to replace NaN values with the current mean value.[1] This data may also have to be preprocessed further in accordance with what may have been done beforehand. This should allow the MLP Classifier the best opportunity to beat the Random Forest Classifier, although it may still not be enough.

Once this has been completed, the MLP Classifier can be trained on the training data and training labels with and without preprocessing and without and without additional data. These results will be output through the MLP Classifier's score function as well as the confusion matrix class and precision_recall_fscore_support class. The resulting data will enable us to analyze and assess the effectiveness of the MLP Classifier. Preprocessing by standardization can then be attempted to see if this improves the performance of the classifier. We can also see how many memorable and non-memorable places there are in the training data here and add more of each to balance out the ratio if necessary. After going through these steps, the MLP Classifier should be at its most effective and the Random Forest classifier can now be worked on for comparison.

2. 2. Random Forest

The approach used for the Random Forest Classifier differ slightly from that seen previously due to the different ways in which they operate. To begin with the process is similar with the importing of the needed libraries and classes which are almost all the same as the MLP Classifier;the only two that differ are the Random Forest Classifier class being used instead of the MLP Classifier class as well as GridSearchCV instead of the Standard Scaler class. This is necessary as they are a different classifier and have a different possible approach to preprocessing. [2]
It is known that pre-processing RF Classifier's datadoes not affect the effectiveness of the classifier and therefore a different approach to preprocessing must be taken. This is by 'preprocessing' the parameters of the classifier itself instead using GridSearchCV to get the best possible parameters for the classifier. This is achieved by first creating and fitting a test RF Classifier and then initializing a parameters dictionary variable that contains a list of estimators and a list of max depths. Using this dictionary as well as the classifier as parameters for the GridSearchCV will allow for the best values for Depth and estimators to be found.[2] These can then be used as the parameters for the actual classifier when it's made and fitted, which would be the next step. Running this new class with the best parameters should allow for the most effective RF classifier possible. Assessment will be possible when the results are outputted in the same way as was done with the MLPClassifier. Looking at this we can then see if adding the additional data in the same way as we did for the MLP Classifier has an affect on performance. If it does allow for a higher precision, recall, score and accuracy then it shall be used. The likely configuration of this will be the same as the MLP Classifier where enough of memorable and non-memorable date are added to balance the ratio. The only issue with adding data is that the GridSearchCV will have to be repeated to get new best values for depth and the number of estimators as the data size would now be larger.[2] Once the best parameters have been found they can then be given to a new classifier when its function is called which can then be fit with the new data. In my opinion, this should produce better overall effectiveness than the MLP Classifier.

3. Results

Now that the classifiers have been appropriately constructed, we can begin to look at how effective each one is, what the best configurations are for them and how they compare to each other.

3. 1.MLP Classifier Results

This will firstly be performed using the MLP Classifier with no pre-processing, pre-processing, no additional data and additional data. With the pre-preprocessing done after additional data has been added. The results for this are as follows;

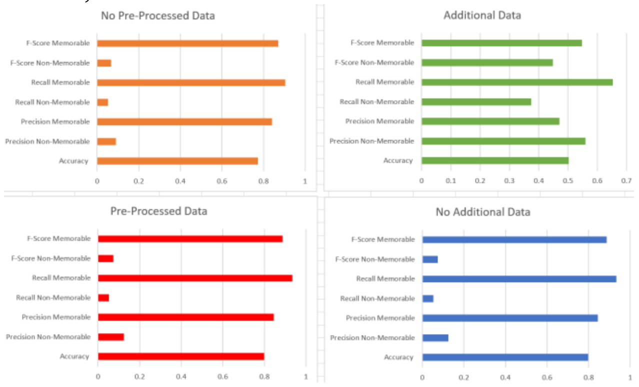

Fig.1 Graphs showing the effects of preprocessing data and adding additional training data to an MLP Classifier to try and improve its performance

It is not clear from this data, however it appears that there is a general improvement when data is pre-processed compared to when it is not. This leads to a 2% increase in accuracy although it is not the best measurement for a classifier, however the other values either stay the same or have a slight increase. For example, Recall of Non-memorable images and the Precision of Memorable stay at about the same whilst the rest of the values increase by about 2%, varying between 1-3%, depending on the category. This means that pre-processing data does improve the outcome, though without additional data non-memorable values are all seen to be very low when compared to memorable values. This is as they sit below 12.5%, but this is seen to improve with additional data balancing the ratio of the training data. This is as it was found that there were more memorable images in the original training data leading to an unbalanced ratio which is rectified with additional data. This leads non-memorable values to become closer to surpassing memorable ones though this can lead to memorable values decreasing. This can be seen in Precision where memorable values go from 90% to 47% whilst non-memorable increases from 9% to 56%. This is the largest change though but score and Recall do follow suit and act in the same way, although not with as large of a change. Accuracy is also seen to decrease but, as stated previously, it is not a very good indication of the performance of a classifier and it is better to rely on the other values. Overall, pre-processing and additional data does allow for a better MLP Classifier;. when the final version was put on Kaggle it got a score of 71.9% which will be used to the compare to RF.

3. 2. Random Forest Classifier Results

TheRF Classifier results were recorded for more data being added and the configuration of the parameters after the data has been added. The results of which are as follows:

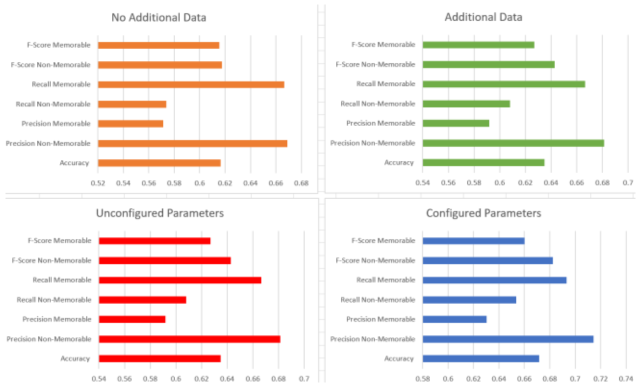

Fig.2 Graphs showing the effects of configuring the parameters and adding additional training data to a RF Classifier to try improving its performance

As seen above, when no additional data is added all values appear to be lower than after the additional data is added. This is as once the additional data is added most categories see an increase of some kind, the only one that does not is memorable image recall which stays the same. All other values increase by 2-3% allowing for a better performing classifier. It should be noted that, both with and without additional data, the RF classifier seems to perform better with non-memorable data than the MLP. However, the MLP seems to hold some advantage when it comes to memorable images as seen in its higher recall and f1-scores.

With regards to configuring the parameters of a RF classifier, many iterations had to be done through the GridSearchCV function to tune the classifier. In total ~80 iterations were performed to achieve the correct parameters, being 32 or 43 estimators and 943 or 1000 as the max depth of the forest. The reason there are two options is due to the Kaggle score which will be explained later, but the ones used for figure 2 are 32 estimators and a max depth of 943. Comparing this to one with a RF Classifier with unconfigured parameters shows a general increase in performance between the two once the correct parameters are applied. This is an increase of about 4% with some values such as accuracy and non-memorable recall increasing by 5%. Though accuracy is not a good indicator of performance, seeing both precision and recall rise at about the same rate to similar values is a good indicator of the classifiers effectiveness. Uploading this classifier and one with different parameters to Kaggle gave interesting results. Although the one featured in the graphs yields better values it does not return a higher score on Kaggle, instead it gives the same score, 74%, as using the other values which returns an interesting prospect. This being that it is possible for two versions of the same classifier to act similarly despite being different at their core.

Comparing this to the MLP Classifier it is clear that the RF Classifier performs better in this task as there is a 3% increase in the Kaggle score. All other values used in the graphs also saw an increase ranging anywhere from 4% in Memorable recall to 30% in non-memorable recall clearly making a Random Forest Classifier more suitable .

References:
[2] https://www.datasciencelearner.com/how-to-improve-accuracy-of-random-forest-classifier/
[1] https://scikit-learn.org/stable/modules/classes.html
[3] https://pandas.pydata.org/pandas-docs/stable/index.html