Candidate Number: 262986

1. Introduction

In this report, our primary goal is to construct a machine learning model that can evolve and classify binary images. These images will be split into being “happy” or “sad” according to the extracted attributes and features. The dataset we are working on is comprised of Convolutional Neural Networks (CNN) and Gist features. The approach we will take in this report consists of preprocessing the given data and choosing the applicable features, then training various classifiers to determine which one fits this task the best for optimal performance.

1.1. Approach

We will begin the classification task by preprocessing the various data. To maintain a balanced representation, we performed stratified sampling to the training data sets. Once complete, we performed a data split into Gist and CNN features while considering the high dimensionality present in the feature space. We also utilized simple and KNN imputation methods to take care of the data that was missing. Feature selection is also an important component of this task and to make the dimensionality smaller to improve the performance of our model we implemented Select Best and correlation analysis methodologies. These analysis techniques allowed us to choose the optimal features that we can later use to train our classifiers with. In order select the correct model for the classification task, we took into consideration many classifiers such as Logistic Regression, Single and Multi Layer Perceptions, Support Vector Machine and Random Forest. We used hyper parameter tuning on the classifier with the best performance before implementing.

2. Methods

We approached this study systematically. First, we loaded the training and testing datasets by utilizing the Pandas library. This ensures that the dataset is easily accessible and structured properly resulting in it being easier to analyze. We then executed stratified sampling on the training data to prevent imbalances, which we later tested the various classifiers on. This helped maintain a representative distribution for our samples in each class of data. [3] Furthermore, we move on to the feature extraction portion where we made use of CNN and Gist features. This gave us a total of 3456 features to work with. The analysis of these features is important to the task as capturing underlying patterns or characteristics is essential. Additionally, we extracted the two labels of “happy” and “sad” along with the confidence labels. This gave us needed information regarding the correctness of assigning the labels. [2]

For feature selection we tried two different techniques. The first is the SelectKBest method which aimed to choose top k best features that have the highest scoring values. We used this along with the chi-squared test. We placed a k value of 100 for each to select the top 100 features that are relevant to the class labels. The second method we conducted was correlation analysis to identify another set of 100 features illustrating high amounts of correlation with the class labels. The idea was that these methods would enhance the power of our model. [4]

For training and testing of our model, a variety of classifiers were used. These include the Single and Multi-Layer Perceptron, Logistic Regression, Support Vector Machine and Random Forest. All of these were trained on the preprocessed training data. Once passed we used cross-validation and checked each classifiers performance on the dataset. Out of all the classifiers, the Random Forest performed best and thus we performed Hyperparameter tuning using HalvingGridSearchCV on it in order to get the optimized parameter values. We were able to obtain the optimal number of estimators, maximum depth and criterion. [6]

The training of the final model was carried out using the scaled training data and optimized hyperparameter on Random Forest. Before passing the data into the classifier we imputed the missing values found in the training data. The imputation consisted of simple and KNN imputation methods resulting in a more finalized, complete dataset. The trained Random Forest Classifier was then employed to predict the class labels in our test data.

3. Results

We will now present our results of our model selection process and discuss the findings that we made. We compared the performance of each classifier we trained on the training data. Here are the results as follows:

Classifier Performance:

- Single Layer Perceptron: Achieved a classification accuracy of 44%.
- Multi-Layer Perceptron: Achieved a classification accuracy of 54%.
- Random Forest: Achieved a classification accuracy of 76.25%.
- Support Vector Machine: Achieved a classification accuracy of 59%.

- Logistic Regression: Achieved a classification accuracy of 52%.

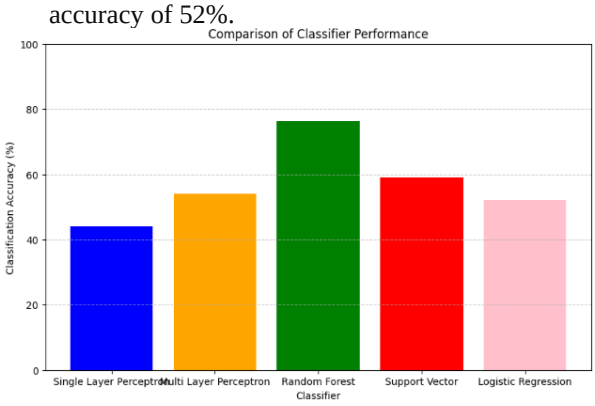

Figure 1: Histogram comparing classifier accuracy.

We performed hyperparameter tuning for the classifier that obtained the highest accuracy on the training data. The results are as followed.

- Number of estimators: [50, 100, 150, 200, 250, 300, 350, 400]
- Maximum depth of the trees: [None, 10]
- Criterion for splitting: ['gini', 'entropy']
- Minimum samples split: [2, 5]
- Maximum features to consider: ['sqrt', 'log2', None]
- Class weight: ['balanced', 'balanced subsample', None]

The best hyperparameters obtained were:

- Number of estimators: 50
- Maximum depth of the trees: 10
- Criterion for splitting: 'entropy'
- Minimum samples split: 2
- Maximum features to consider: 'log2'
- Class weight: 'balanced subsample'

Furthermore, we used 5-fold cross validation to evaluate the performance of the final classifier. [6] The classification accuracy scores for each fold were as follows:

1. Fold 1: 72.5%
2. Fold 2: 66.25%
3. Fold 3: 76.25%
4. Fold 4: 78.75%
5. Fold 5: 77.5%

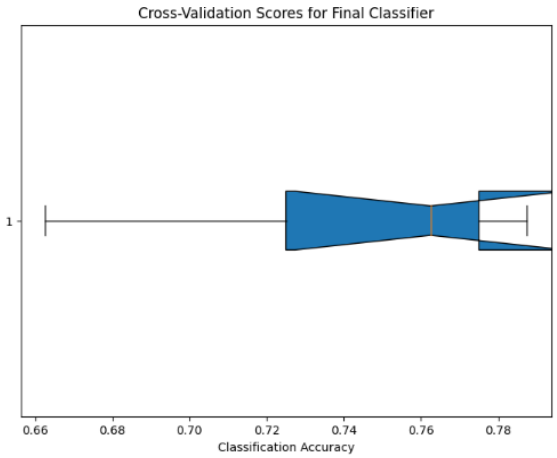

Figure 2: Box Plot of classifier performance

3.1. Discussion

Through hyper parameter tuning we were able to find the most efficient set of parameters and apply them to our chose classifier of Random Forest. This led us to an improved classification accuracy. The comparison of the performance of classifiers demonstrated that Random Forest was the best fit for our training data, followed by Support vector machine then Multi-Layer Perceptron, then Logistic Regression and Finally the Single Layer Perceptron. In Figure 2 we can see the cross-validation scores of the final classifier through five different folds. The scores fell within a small range which illustrates the classifier is robust being able to generalize the data properly. [1]

4. Conclusion

In this Study we made use of different classifiers to come up with our own optimal classifier for a binary classification task on a set of data containing images that were either “happy” or “sad”. From the results and data, we can conclude that the Random Forest classifier is the best for the task at hand. In the future we can further tune the hyper parameters which may allow for an even more optimal output of the binary classification task given.

4.1. References

References

[1] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[2] Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

[3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011).

[4] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

[5] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial intelligence, 97(1-2), 273-324.

[6] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. springer.