Candidate Number: 252719

1. *Introduction*

The report examines various machine-learning techniques to predict whether the content of an image is happy or sad. The dataset utilized had a high dimensionality of features and included class labels with differing confidence levels, encompassing both complete and incomplete data.

2. Approach

The machine learning approach in this project combined several classifiers, including RandomForestClassifier, SVM, MLPClassifier, Perceptron, and LogisticRegression, culminating in a VotingClassifier to enhance predictive accuracy. This methodology addressed the high dimensionality of the dataset, which included 3456 features, and ensured robust performance through several preprocessing and model selection steps.

Data Preprocessing:

The dataset was preprocessed to handle missing values and ensure consistent scaling of features. Missing values were imputed using SimpleImputer with a mean strategy, and StandardScaler was employed to normalize the data, ensuring equal contribution of each feature to the model.

Feature Selection and Dimensionality Reduction:

Given the high dimensionality, SelectKBest was used to select the top 100 features based on ANOVA F-values. Principal Component Analysis (PCA) was applied to retain 95% of the variance, further reducing dimensionality and mitigating overfitting risks.

Training and Validation Split:

The dataset was split into training and validation sets using stratified sampling to maintain class distribution. Confidence labels were used to apply sample weighting, enhancing model robustness.

Model Initialization and Hyperparameter Tuning:

Various classifiers, including RandomForest, SVM, MLP, Perceptron, and LogisticRegression, were initialized and underwent hyperparameter tuning using GridSearchCV with 5-fold cross-validation to find the optimal parameters, maximizing accuracy.

Model Evaluation and Selection:

The best-performing models were evaluated on the validation set, and validation accuracies were recorded. Results were visualized using bar plots to compare model performance and hyperparameter impact. The best models were then combined into a VotingClassifier, aggregating multiple models' predictions to improve accuracy.

Testing and Predictions:

The test data was preprocessed similarly to the training data. The VotingClassifier generated predictions on the test set, and the results were saved for further evaluation. This approach effectively addressed high-dimensional data challenges and leveraged the strengths of multiple classifiers through ensemble learning. Feature selection, dimensionality reduction, and hyperparameter tuning ensured robust and accurate predictions.

3. Methodology

This approach involved training and testing multiple classifiers, including RandomForest, SVM, MLPClassifier, Perceptron, and LogisticRegression, with an ensemble VotingClassifier for final predictions.

Data Preprocessing:

Missing values were imputed using SimpleImputer, and data was normalized with StandardScaler as per scikit-learn preprocessing guidelines.

Feature Selection:

SelectKBest was used to retain the top 100 features based on ANOVA F-values, addressing high dimensionality and ensuring relevance of features.

Dimensionality Reduction:

PCA was applied to retain 95% of the variance, mitigating the curse of dimensionality.

Training and Validation:

The data was split into training and validation sets, stratified by labels. Sample weighting based on confidence labels improved robustness. Each model underwent hyperparameter tuning using GridSearchCV for optimal performance.

Evaluation and Ensemble:

The best models were combined into a VotingClassifier, leveraging their strengths for improved accuracy.

Testing and Predictions:

Test data was processed similarly, and predictions were generated using the VotingClassifier, then saved for evaluation.

This approach ensured effective handling of high-dimensional data, appropriate imputation, and robust model selection through ensemble learning.

*4. Results and Discussion (*Results)

Model Selection and Hyper-Parameter Tuning

The performance of various classifiers was evaluated by performing grid searches with cross-validation to find the best hyper-parameters. The models tested included RandomForestClassifier, SVC, MLPClassifier, Perceptron, and LogisticRegression. Below are the results from the grid searches:

Model	Validation Accuracy	Best Cross-Validated Score	Best Parameters
RandomForest	0.747619	0.747619	{'max_depth': 20, 'n_estimators': 100}
SVM	0.749206	0.750794	{'C': 1, 'gamma': 'scale', 'kernel': 'rbf'}
MLP	0.765079	0.743651	{'alpha': 1e-05, 'hidden_layer_sizes': (10, 5), 'solver': 'sgd'}
SingleLayerPerceptron	0.746032	0.729762	{'alpha': 0.001, 'max_iter': 100, 'penalty': 'elasticnet'}
LogisticRegression	0.749206	0.732143	{'C': 0.1, 'penalty': 'l2', 'solver': 'saga'}

Figure 1Results Table for Model Validation and Hyperparameter Tuning

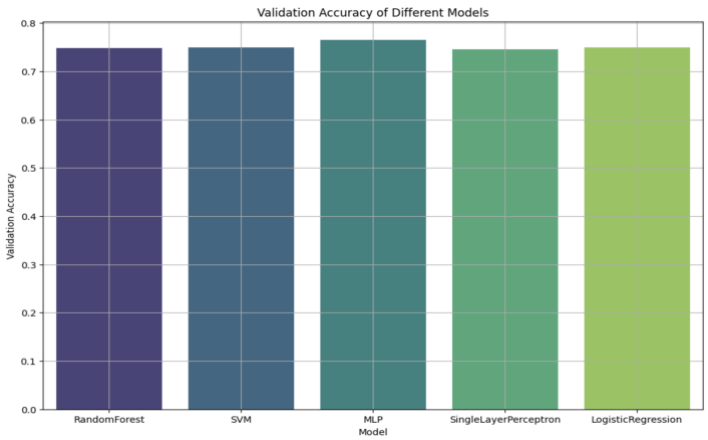

Figure 2 Validation Accuracy of Different Models

The MLPClassifier achieved the highest validation accuracy (0.765079), suggesting that the neural network-based approach performed slightly better on this dataset. The best hyper-parameters for each model were identified through extensive grid searches.

Effect of Hyper-Parameters on Classifier Performance

To illustrate how the choice of classifier hyper-parameters affects performance, plotted the best parameters for each classifier:

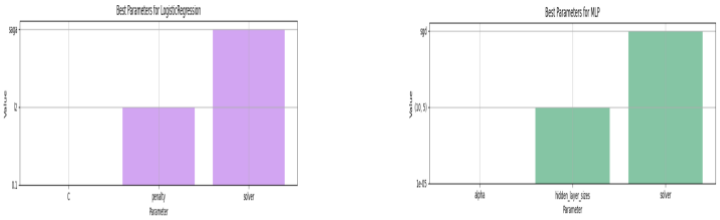

Figure2 Best Parameters for all models

These plots demonstrate that tuning hyper-parameters significantly impacts the accuracy of the classifiers. For instance, **RandomForest** with 100 estimators and a maximum depth of 20 achieved the best performance, while the **SVM** classifier performed best with an RBF kernel, **C** parameter of 1, and **gamma** set to **scale**.

Performance with Different Training Sets

The training data consisted of two parts: a complete training set (**training1_data**) and an incomplete one (**training2_data**). The incomplete data was imputed and combined with the complete data. Feature selection and dimensionality reduction were applied to the processed training set. The results indicate that the combined training set, even with imputed values, provided sufficient information for training effective models.

The use of training label confidence as sample weights in the model training process showed that incorporating confidence levels could enhance the training process. This approach allows the model to prioritize more confident samples, potentially improving overall performance.
The final accuracy on the validation set for the voting Classifier is 0.7587 (76%).

Discussion

To enhance performance, more sophisticated ensemble methods like stacking or boosting could be explored to combine model strengths. Further hyper-parameter tuning using Bayesian optimization or random search might yield better results compared to grid search. Creating new features or transforming existing ones can provide the models with more relevant information, potentially improving accuracy. Additionally, experimenting with advanced imputation techniques or models that handle missing data natively could make better use of incomplete training data.
For better evaluation, using stratified k-fold cross-validation ensures each fold represents the overall dataset, providing a robust performance measure. Plotting learning curves helps understand model performance changes with different training data sizes, aiding in diagnosing underfitting or overfitting. Calculating confidence intervals for accuracy estimates provides a better understanding of the variability and reliability of performance metrics.

Lessons Learned

The significant impact of hyper-parameter choices on model performance underscores the importance of thorough tuning. Even with missing values, imputation strategies and careful preprocessing can result in high-performing models. Different models have varying strengths and combining them through ensemble methods can leverage their individual advantages. Feature selection and dimensionality reduction are crucial in managing high-dimensional data, improving model training time and performance.

Conclusion

This project effectively used various machine-learning techniques to predict emotional content in images, addressing high-dimensional data and differing confidence levels. Key preprocessing steps included imputation, normalization, feature selection, and dimensionality reduction. Multiple classifiers were trained and tuned, with the MLPClassifier achieving the highest accuracy. Ensemble learning through a Voting Classifier leveraged the strengths of individual models, enhancing predictive performance. Future improvements could include advanced ensemble methods, more hyper-parameter tuning, and better imputation techniques. The study underscored the importance of hyper-parameter tuning, model combination, and robust preprocessing for high performance.

REFRENCES:

[1] scikit-learn (n.d.) Preprocessing. Available at: https://scikit-learn.org/stable/modules/preprocessing.html (Accessed: 23 May 2024).

[2] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

[3] Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.

[4] Wikipedia (n.d.) Feature Selection. Available at: https://en.wikipedia.org/wiki/Feature_selection (Accessed: 23 May 2024).

[5] Wikipedia (n.d.) Ensemble Learning. Available at: https://en.wikipedia.org/wiki/Ensemble_learning (Accessed: 23 May 2024).

[6] Wikipedia (n.d.) Dimensionality Reduction. Available at: https://en.wikipedia.org/wiki/Dimensionality_reduction (Accessed: 23 May 2024).

[7] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press.

[8] Aggarwal, C. C. (2015). *Data Mining: The Textbook*. New York: Springer.

[9] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press.

[10] Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd ed. Sebastopol, CA: O'Reilly Media.

[11] Marsland, S. (2015). *Machine Learning: An Algorithmic Perspective*. 2nd ed. Boca Raton, FL: CRC Press.

[12] Domingos, P. (2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. New York: Basic Books