MACHINE LEARNING REPORT

Hailing Cai
Department of Computer Science
Durham University
llpw83@durham.ac.uk

1 Comprehensive Dataset Exploration

1.1 KNOW

The dataset contains **561 brain features**, **100 image features**, and **512 text features** for each sample. The number of features is consistent across all samples. The category distribution is balanced, with **10 samples per label** for the seen category and **80 samples per label** for the unseen category.

1.2 SEE

Histograms visualise the distribution of brain, image and text features. The distributions of brain and text features are relatively symmetric, but **75% of image features exhibit significant skewness (absolute skewness > 0.5)**, indicating that the image data distribution is severely asymmetric and may significantly impact the performance of models based on the normality assumption.

Heat map analysis showed that: the correlation between **brain and image features** was strongest in the first **20%**, and then weakened, indicating that specific image features were related to brain area activities; **brain and text features** were weakly correlated as a whole; and **images were weakly and negatively correlated with text features**, reflecting the independence of information expression.

Histograms and KDE curves were analysed for feature-label correlations. **Brain features** were weakly correlated (**-0.02 to 0.03**) and right-skewed, **image features** were more widely distributed and symmetrical (**-0.06 to 0.04**), and **text features** had the widest range (**-0.06 to 0.08**) and were right-skewed. Overall, all features have a weak linear relationship with the labels, with correlation coefficients close to zero.

The **Z-score and IQR methods** were used for outlier detection for all three features.The Z-score method marked data points with an absolute Z-score greater than 3 as outliers, while the IQR method considered data points exceeding 1.5 times the interquartile range as outliers. The results show that **each feature contains outliers**.

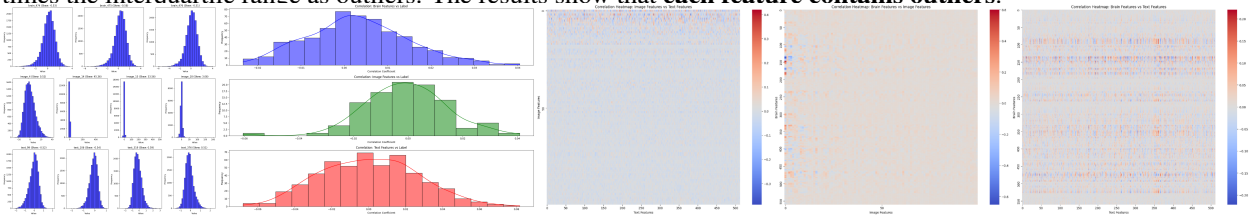

Figure 1: Caption

1.3 Find

The dataset analysis shows that **seen classes** have fewer samples per label than **unseen classes**. This distribution will limit the model's learning of seen classes and affect its generalization ability. Additionally, the **skewed distribution of image features** creates an imbalance of information between classes, which may cause **bias in model predictions** for specific categories.

The weak correlations between features may lead to **information redundancy or feature interference**, affecting the model's stability. Furthermore, the presence of outliers in the data not only increases training difficulty but also **causes**

optimization issues, such as abnormal gradient updates, slower convergence, and excessive sensitivity to extreme samples.

Based on these data characteristics, the Random Forest model was chosen, as its multi-tree structure can not only effectively handle complex nonlinear feature spaces, but also improve the processing capability for skewed distribution data through diverse data partitioning methods, while reducing overfitting risk to enhance model robustness.

2 Custom Model Implementation

2.1 IMPLEMENT

The decision tree is constructed based on Gini impurity, selecting the optimal feature through recursive data splitting. The tree growth is constrained by the maximum depth to prevent overfitting. The random forest ensemble consists of 20 decision trees, with each tree using Bootstrap sampling and random feature selection for splits, ultimately producing predictions through majority voting.

During the training process, **PCA and UMAP** dimensionality reduction are used to process three modal data types, which are then fused into a **unified feature matrix**. `RandomForest.fit()` performs **sampling with replacement** through **Bootstrap sampling** (`sample_size=100`) to ensure tree diversity. The model contains **20 decision trees** (`n_estimators=20`), with each tree using `DecisionTree.fit()` to recursively partition data based on **Gini impurity**. The model records **training time and validation accuracy** to monitor the balance between performance and computational cost.

During the **testing process**, `RandomForest.predict()` makes independent predictions through multiple decision trees and uses **majority voting** to determine the final result. **Test set accuracy** is calculated to evaluate generalization performance, and **inference time** is recorded to measure computational efficiency. The relationship between **training time** and **validation accuracy** versus number of trees is visualized.

2.2 COMPARE

Compared to the baseline model, the training time of **my implemented model** is significantly higher, exceeding 500 seconds, but the **baseline model** takes only 0.22 seconds to train, highlighting the huge gap in computational efficiency. In terms of convergence performance, the **baseline model** shows greater **stability**, maintaining an accuracy level of 0.95 after 10 trees, while **my model** shows significant fluctuations, suggesting that the convergence process is unstable and requires further improvement. This difference in performance may be due to a more optimised implementation of the baseline model with a more efficient computational strategy.

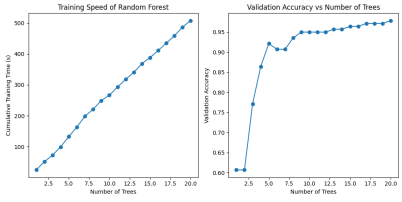

Figure 2: my model

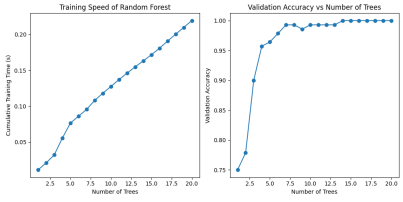

Figure 3: baseline model

Figure 4: convergence performance and training time

2.3 IMPROVE

Three main optimisations: Feature filtering (**150 important features** selected by the SHAP method), parameter tuning (**increased to 40 decision trees**), and algorithm improvement (**optimised best_split()** function). After optimisation, the training time is reduced to **123.00 seconds**, the testing accuracy reaches **88.33%**, the validation accuracy reaches **95%**, and the feature processing speed is improved by **3 times**.

3 Result Analysis and Visualisation

3.1 PERFORMANCE

Table 1: Comparison of Model Performance Metrics

	Model	Baseline	Improved
Training Time (s)	507.82	0.37	123.00
Accuracy	83.33%	83.33%	88.33%
Precision	0.8383	0.8750	0.8667
Recall	0.8333	0.8333	0.8833
F1-score	0.8080	0.8298	0.8681

Training Time: The baseline model uses an optimised library for fast training. The initial version is computationally complex resulting in slow training. The optimised model reduces training time by **75%**and allows the number of trees to be increased to improve accuracy.

Accuracy: The initial model achieves the classification level of the benchmark library model, proving that the Gini splitting, recursive construction and integrated voting mechanism are correctly implemented. The 5% improvement in model accuracy after optimisation is attributed to the improved robustness of the feature filtering and splitting strategy optimisation.

Precision: Baseline accuracy is low and the initial model is boosted, showing good misclassification control. This validates the effectiveness of the algorithm implementation and decision tree construction strategy. The optimised version has a slight decrease in precision, possibly due to an increase in minor misclassification as a result of improved recall.

Recall: The baseline model and the initial model have similar recall. The optimised version significantly improves recall and category recognition by improving the `best_split()` function and increasing the number of decision trees.

F1 Score: The baseline model has a low F1 value, while the initial model achieves a higher score. The F1 score is significantly improved after optimisation, indicating that the model enhances the recognition rate while reducing misclassification.

3.2 VISUALISATION

The **optimised model** demonstrated significant performance improvements, where the number of correct classifications on the diagonal **increased to** 15 from 12 in the baseline model and 14 in the initial version of the model, confirming the effective enhancement of the model's learning capability. However, the analyses show that the **Label 11 still suffers from recognition difficulties**, which may be attributed to the high feature overlap between this category and the other categories. Additionally, the data shows that the number of **correct classifications for four categories remains at 2**, which indicates that the feature differentiation ability of specific categories still needs to be improved, possibly due to the lack of clear definition of the decision boundary or the optimisation space of the `best_split()` algorithm has not yet been fully released. It is recommended to improve the model's recognition accuracy for these challenging categories by optimising the feature selection mechanism and adjusting the model parameters.

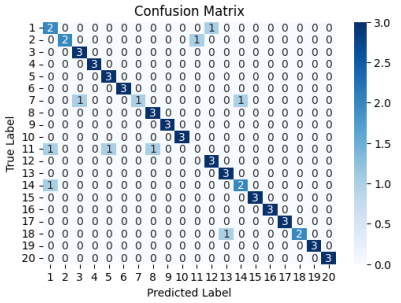

Figure 5: initial model

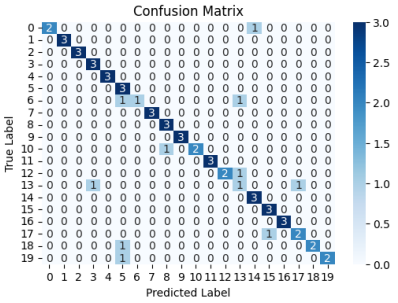

Figure 6: Baseline model

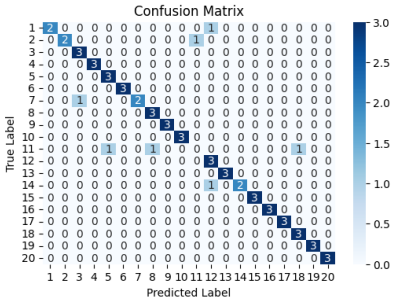

Figure 7: optimized model

3.3 ABLATION

Experiments show that removing `best_split()` optimisation or feature filtering increases training time, while the fully optimised model trains the fastest, confirming that these optimisations effectively reduce computational overhead.

Feature filtering contributes the most to model accuracy, with features selected by the SHAP method significantly improving performance. `best_split()` optimisation improves precision and recall. Overall, the fully optimised model performs best in all metrics and successfully solves the previously identified problems of feature relevance, computational efficiency, and category differentiation.

4 Paradigm Design and Data Splitting

4.1 PARADIGM

The data division strategy was adjusted from $7:3$ (70% training data, 30% test data) to $5:5$ (50% training data, 50% test data). The results show an increase in training time ($123.00s \rightarrow 145.90s$) and a decrease in test accuracy ($88.33\% \rightarrow 81.00\%$). This change has the following main effects:

1. The decrease in training data leads to insufficient feature learning and affects the model generalisation ability.
2. Reduced feature dimensionality after PCA and UMAP dimensionality reduction, which may limit the model expressive power.
3. Decision trees require deeper structures to capture complex patterns, increasing computational effort.
4. Bootstrap sampling and SHAP feature selection are less stable and require additional computation time.

4.1.1 Practical Value

Although the training time increases, it reduces the data collection and labelling costs, while improving the representativeness of the test set, making the evaluation results more stable and reliable.

4.2 ADJUSTMENT

4.2.1 Adaptation to Reduced Generalisation Ability

In order to adapt to the problem of **reduced generalisation ability** due to the **reduced training samples** in the $5:5$ data partition, the **dimensionality of the feature selection** was increased to ensure that the model can extract more key information to compensate for the **loss of information** due to the reduced training data.

In data preprocessing, we use **SHAP** to filter out the **important features** before using **PCA** and **UMAP** for dimensionality reduction to ensure that the computational complexity is reduced and dimensional redundancy is minimised while maximising the retention of key information. To achieve the effect of **enhancing the data representation ability**, so that the model still maintains **high learning ability** and **generalisation performance** in the case of reduced training data.

4.3 REFLECTION

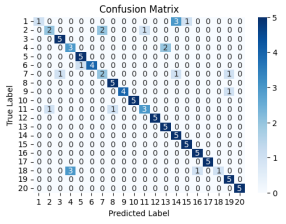

Figure 8: *Visualization of Model Performance*

	New Dataset Splitting	Original
Training Time (s)	170.94	123.00
Accuracy (%)	80.00	88.33
Precision	0.8284	0.8667
Recall	0.8000	0.8833
F1-score	0.7764	0.8681

Figure 9: *Comparison of Model Performance*

Although the hyperparameters were adjusted and data preprocessing was optimised to better increase **dimensional information**, the 5:5 data division still resulted in a **decrease** in model performance compared to 7:3. As an integrated learning method based on multiple decision trees, **Random Forest** requires **sufficient training data** to learn stable decision boundaries, and the reduction in the amount of training data resulted in the model not being able to adequately capture the pattern of category distribution. This is demonstrated by an increase in **computation time** (**170.94s vs. 123.00s**), an **8% decrease in accuracy**, a decrease in both **precision and recall**, and a decrease in the **F1-score from 0.8681 to 0.7764**, suggesting that while the 5:5 division increases the **test dataset** to make the evaluation more representative, it also restricts the model's ability to learn. However, **Label 11's** classification accuracy instead improved from 0 to 3 (out of a total of 5), which may be due to the **larger test dataset** providing the opportunity for a more comprehensive category evaluation.