Machine Learning Prediction of Patients with schizophrenia from Anatomical Brain Imaging

Candidate Number: HVWM7

COMP0189 - Applied Artificial Intelligence

1 Introduction

Schizophrenia is a brain disorder characterized by neurocognitive deficits and structural brain abnormalities. Patients with schizophrenia exhibit progressive brain atrophy and reduction in the thickness of the cerebral cortex, which can influence the Total Intracranial Volume (TIV), the Cerebrospinal Fluid (CSF) and indirectly the White Matter (WM) Volume. Moreover, the cerebral cortex is composed predominantly of Gray Matter (GM), which contains the cell bodies of neurons. Cortical thinning directly reflects a reduction in GM volume in affected areas of the brain, as it involves the loss or shrinkage of neuronal tissue in the cortex [4] [2] [1] [1] [3]. Brain volume loss has shown to be more pronounced in the early stages of the disease and is associated with clinical outcomes [5].

This study aims at predicting patients with schizophrenia from their anatomical brain images: it takes a set of 410 training patients and 103 testing patients, all evaluated from two Voxel-based-morphometry (VBM): **Region of Interest (ROIs)**, based on the GM scaled for the TIV, with `284` brain ROIs, and whole **VBM** image of the brain, based on the affine transformation to MNI referential, with `331695` grey matter voxels. The purpose of this project is to draw a binary classification of patients with or without Schizophrenia. This will be conducted with three types of machine learning models: linear, non-linear and tree-based, on both datasets, as long as computation power allows it. Moreover, a performance comparison with stratified group methods is conducted. Thus, the objective can be divided into three parts: exploring different feature sets, evaluating various models, and experimenting with cross-validation strategies

2 Methods

First and foremost, the complexity of having two datasets, one with a high feature-on-sample ratio, calls for a dataset-focused approach. Following the scikit-learn's algorithm cheat sheet, as it provides a key Chain of Thought for selecting a model, it was chosen to work with the following models: `LinearSVC`, `RandomForestClassifier` and `KNN`. However, to satisfy research interests and pure curiosity, the decision was made to extend the analysis by including three other models: `LogisticRegression`, `XGBoost` and `SVM` with non-linear kernels. *The main three models will be displayed in bold to keep the focus on the original choice.* Not all models were run on the VBM dataset, because of their high computational needs. Table 1 summarises the possible models for each dataset, further context and reflection are given in the Results and Discussion phases.

Two pipelines were experimented with: first a baseline of the selected models, then a more thorough approach to optimise with hyperparameters, regularisation and feature engineering.

The first pipeline focused on conducting baseline experiments with the provided models and running them with their standard parameters to have a first assessment of their computation needs and results. The pipeline is available in the Algorithm 1. The results are provided in Table 2 for the ROI dataset, and Table 9 for the VBM dataset. This first pipeline allowed to evaluate on the bAcc and ROC-AUC, with a dataframe outputted for each model: the results comments are available on the Results section

The second pipeline focused on upgrading this baseline with more hyperparameter tuning, regularisation and feature engineering techniques to ensure better results. First, a `StandardScaler` preprocessing step was added to the pipeline, as it's more robust to outliers, which is particularly relevant with imaging data, meaning that outliers do not significantly skew the mean and standard deviation of the data. Then, the `y_train` and `y_test` sets were encoded with `LabelEncoder()` in order to compute key new metrics, with the final list of metrics being: **ROC-AUC**, **precision**, **recall**, **f1**, **fit time**. For some models, a precomputed kernel was used in order reduce the computational cost. Finally, a Grid Search is implemented for hyperparameter tuning, including regularisation hyperparameters. The detailed steps are available in the Algorithm 2. The results are provided for each model, comparing for the ROI dataset in Table 2, and Table 9 for the VBM dataset. This first pipeline allowed to evaluate on the bAcc and ROC-AUC, with a dataframe outputted for each model: the results comments are available on the the Results section.

Both pipelines use cross-validation techniques with both `StratifiedKFold` and `StratifiedGroupKFold`,to ensure that the models were trained and validated on different subsets of the data, providing a robust evaluation of their performance. The groups were set from the participants_train dataset, with $groups = np.array(participants\_train['site\_encoded'].astype("int"))$. For representativity reasons, tt was chosen to group by site instead of by sex: while gender is a key feature when it comes to creating a balanced dataset, StratifiedGroupKFold remains key for getting "closer" to iid data. We would like to know if a model trained on a particular set of groups generalizes well to the unseen groups: to measure this, we need to ensure that all the samples in the validation fold come from groups that are not represented at all in the paired training fold." Among the features in `participants_train`, the `site` was the most relevant.

As we are working on a binary classification problem, a few additional metrics were selected to provide a more comprehensive performance assessment and complement the already present balanced accuracy (bAcc) and (ROC-AUC): **precision** helps assess the model's ability to avoid false positives, which is critical in schizophrenia detection, **recall** evaluates the model's ability to identify all positive instances, while the **f1-score** provides a balanced measure that considers both metrics. Additionally, the **fit time metric** was added to assess the computational need and training efficiency of the model. This is particularly important when dealing with large datasets like the VBM dataset, especially in our resource-constrained environments. All models were run on a CPU to ensure a fair comparison.

3 Results

An overview of the models implemented for each dataset is presented Table 1. For the ROI dataset, all models were tested, while for the VBM dataset, only `RandomForestClassifier` was applied due to computational constraints.

The baseline results for the ROI dataset are presented in Table 2. The tree-based models, particularly `XGBoost`, achieved the highest performance, with a bACC of 0.75 and ROC-AUC of 0.84 using both cross-validation strategies. The linear models (SVC with linear kernel and LogisticRegression) also performed well, with bACC ranging from 0.71 to 0.72 and ROC-AUC from 0.80 to 0.82. The non-linear models (`KNN` and `SVC with RBF kernel`) had lower performance compared to the other model families.

Tables 3 to 8 show the detailed results for each model applied to the ROI dataset, comparing the baseline performance with the results obtained after hyperparameter tuning using GridSearch. The GridSearch generally led to improvements in performance metrics, with the largest gains observed for the `SVC with the linear kernel` (Table 3) and `KNN` models. `Logistic Regression` (Table 4), on the other hand showed limited post-tuning enhancements post-tuning are modest. The `RandomForestClassifier`, as detailed in Table 5, shows notable improvements in recall, but the slight decrease in precision suggests a trade-off, where the model might be more prone to false positives after tuning. While `XGBoost` is a powerful and versatile model (Table 6), its hyperparameters must be carefully managed to avoid fitting the model too closely to the training data, which can harm its generalization to new data. The `KNN` model - Table 7, benefits significantly from hyperparameter tuning, with improvements in balanced accuracy, ROC-AUC, and recall. These enhancements come with an increased computational burden, highlighting the resource-intensive nature of KNN, especially as dataset size and dimensionality grow, making it impossible for VBM. Lastly, the `SVC with RBF kernel`, shown in Table 8, demonstrates substantial performance improvements post-tuning.

For the VBM dataset, the baseline results are shown in Table 9.

`RandomForestClassifier` achieved a bACC of 0.63 and ROC-AUC of 0.74 using `StratifiedKFold`, while the `SVM model` obtained a bACC of 0.65 and ROC-AUC of 0.70. However, the computational time for the `SVM model` was significantly higher (1670 seconds) compared to `RandomForestClassifier` (37 seconds). Table 10 presents the detailed results for `RandomForestClassifier` applied to the VBM dataset. The GridSearch led to a slight decrease in performance compared to the baseline, possibly due to overfitting or the limited hyperparameter search space.

The `RandomForest` model was submitted on the dedicated RAMP challenge. All results are available in the Results section.

4 Discussion

The analysis of the results shows that the tree-based models, particularly `XGBoost` and `RandomForestClassifier`, consistently performed well across both datasets. The linear models also showed good performance on the ROI dataset, while the non-linear models had lower performance. The computational cost was generally higher for the VBM dataset due to its high dimensionality. The GridSearch approach for hyperparameter tuning led to improvements in performance for most models applied to the ROI dataset, but not for the VBM dataset. The linear models, while slightly less performant than tree-based models, offered a commendable balance between performance and computational cost, making them viable options for scenarios where computational resources are limited. However, in the high-dimensional VBM dataset, the `RandomForestClassifier` stood out not only for its relatively high performance but also for its computational efficiency, a critical factor given the computational constraints associated with high-dimensional data. The `SVM model`, despite its slightly higher balanced accuracy and ROC-AUC in the VBM dataset, was markedly less efficient, with a computational time significantly higher than that of the `RandomForestClassifier`, underscoring the challenges and trade-offs inherent in managing high-dimensional data. While `StratifiedKFold` might be preferable for achieving stable and generalizable performance estimates, `StratifiedGroupKFold` provides a more stringent test of the model's generalization capability, which is invaluable in clinical settings where models must perform reliably across diverse patient populations. Finally, tree-based models, with their blend of high performance and computational efficiency, are particularly suited for clinical applications in both low and high-dimensional settings, provided that the choice of cross-validation strategy aligns with the specific requirements and constraints of the application domain.

5 Conclusion

In conclusion, this study aimed to predict schizophrenia from brain MRI, using machine learning models applied to two datasets. The results showed that tree-based models, particularly `XGBoost` and `RandomForestClassifier`, consistently performed well across both datasets, while linear models offered a good balance between performance and computational cost. The study highlights the importance of considering computational efficiency, the impact of cross-validation strategies, and the dataset-focused model selection. Future work could involve conducting more rigorous statistical tests, both for a better feature importance assessment and to further validate the significance of the results. Moreover, an interesting extension could be to combine feature importance analysis - with the `participants` dataset, with domain knowledge to create a more interpretable and clinically relevant feature set for the models.

Appendix

1 Algorithms

Algorithm 1 Baseline Evaluation

Require: $model, cv, f\_extractor, X\_train, y\_train, X\_test, y\_test, groups$

Ensure: $result\_df$

Setup:

$estimator \leftarrow make\_pipeline(f\_extractor, model)$

▷ Setup pipeline with feature extractor and model

Cross-Validation:

$scoring \leftarrow [\text{"}balanced\_accuracy\text{"}, \text{"}roc\_auc\text{"}]$

$cv\_results \leftarrow cross\_validate(estimator, X\_train,$
$y\_train, scoring, cv, return\_train\_score = True, groups)$

▷ Perform cross-validation

Baseline Scores:

$baseline\_bacc \leftarrow cv\_results[\text{"}test\_balanced\_accuracy\text{"}].mean()$

$baseline\_auc \leftarrow cv\_results[\text{"}test\_roc\_auc\text{"}].mean()$

Refit and Test:

$estimator.fit(X\_train, y\_train)$

$y\_pred\_test \leftarrow estimator.predict(X\_test)$

$score\_pred\_test \leftarrow estimator.predict\_proba(X\_test)[:, 1]$

$test\_bacc \leftarrow balanced\_accuracy\_score(y\_test, y\_pred\_test)$

$test\_auc \leftarrow roc\_auc\_score(y\_test, score\_pred\_test)$

Results:

$results \leftarrow \{\text{"}CV\,BalancedAccuracy\text{"} : baseline\_bacc, \text{"}CV\,ROC - AUC\text{"} : baseline\_auc,$
"Test Balanced Accuracy": test_bacc, "Test ROC-AUC": test_auc}

$result\_df \leftarrow DataFrame(results, index = [0])$

return $result\_df$

Algorithm 2 Evaluate Model with Grid Search and Multiple Metrics Including Fitting Times

Require: $model, param\_grid, cv, f\_extractor, X\_train, y\_train, X\_test, y\_test, groups$
Ensure: $result\_df$

Setup Pipeline:
$estimator \leftarrow make\_pipeline(f\_extractor, StandardScaler(), model)$
Initial Cross-Validation:
$cv\_results \leftarrow cross\_validate(estimator, X\_train, y\_train, scoring = \{"balanced\_accuracy", "roc\_auc", "precision", "recall", "f1"\}, cv = cv, return\_train\_score = True, return\_estimator = True, groups = groups)$
▷ Evaluate baseline metrics
$baseline\_metrics \leftarrow$ Extract mean scores from $cv\_results$
$baseline\_fit\_time \leftarrow$ Average fitting time from $cv\_results$
Grid Search Setup:
$grid\_search \leftarrow GridSearchCV(estimator, param\_grid, scoring = "balanced\_accuracy", refit = "balanced\_accuracy", cv = cv, verbose = 1, n\_jobs = -1)$ ▷ Configure Grid Search
Grid Search Execution:
$grid\_search.fit(X\_train, y\_train, groups = groups)$ ▷ Find best parameters
$best\_model \leftarrow grid\_search.best\_estimator\_$
$grid\_search\_fit\_time \leftarrow$ Mean fitting time from $grid\_search.cv\_results$
Best Model Evaluation:
$y\_pred\_test \leftarrow best\_model.predict(X\_test)$
$score\_pred\_test \leftarrow best\_model.predict\_proba(X\_test)[:, 1]$
Calculate test metrics: balanced accuracy, ROC-AUC, precision, recall, f1
Result Compilation:
$result\_df \leftarrow DataFrame$ with baseline, GridSearch metrics, and differences
return $result\_df$

2 Results

Model Family	Model	*ROI Dataset*	*VBM Dataset*
Linear	**SVC, linear kernel**	✓	✗
	LogisticRegression	✓	✗
Tree-based	**RandomForestClassifier**	✓	✓
	XGBoost	✓	✗
Non-linear	**KNN**	✓	✗
	SVC, non-linear kernels	✓	✗

Table 1. Models implementations per dataset

ROI Dataset

ROI Results							
Model Family	Model	**StratifyKFold**			**StratifyGroupKFold**		
		bACC	ROC-AUC	Time	bACC	ROC-AUC	Time
Baseline	*MLP*	*0.63*	*0.74*	-	0.71	0.8	2.7
Linear	**SVC, linear kernel**	0.71	0.8	6.45	0.71	0.8	3s
	Logistic Regression	0.72	0.82	16s	0.72	0.82	8s
Tree-based	**RandomForestClassifier**	0.73	0.80	9s	0.73	0.80	2s
	XGBoost	0.75	0.84	5s	0.75	0.84	2
Non-linear	**KNN**	0.63	0.68	7s	0.63	0.68	2
	SVC, RBF kernel	0.64	0.68	5s	0.64	0.68	2s

Table 2. Baseline models results for the ROI dataset

	StratifiedKFold			StratifiedGroupKFold			*GridSearch Difference*
Metric	Baseline	GridSearch	*Difference*	Baseline	GridSearch	*Difference*	
Balanced Accuracy	0.68	0.74	*0.07*	0.62	0.72	*0.12*	*0.02*
ROC-AUC	0.73	0.83	*0.10*	0.69	0.82	*0.14*	*0.01*
Precision	0.67	0.76	*0.10*	0.61	0.75	*0.16*	*0.01*
Recall	0.61	0.67	*0.06*	0.63	0.67	*0.04*	*0.00*
F1 Score	0.64	0.71	*0.08*	0.61	0.70	*0.10*	*0.01*
Fit Time (s)	1.45	7.18	*5.72*	1.62	5.73	*4.11*	*1.45*

Table 3. SVC, kernel=linear

	StratifiedKFold			StratifiedGroupKFold			*GridSearch Difference*
Metric	Baseline	GridSearch	*Difference*	Baseline	GridSearch	*Difference*	
Balanced Accuracy	0.73	0.75	*0.02*	0.71	0.74	*0.04*	*0.01*
ROC-AUC	0.81	0.84	*0.03*	0.80	0.83	*0.04*	*0.01*
Precision	0.73	0.79	*0.06*	0.72	0.79	*0.07*	*0.00*
Recall	0.68	0.65	*-0.03*	0.66	0.66	*-0.02*	*-0.01*
F1 Score	0.70	0.71	*0.01*	0.68	0.71	*0.03*	*0.00*
Fit Time (s)	1.41	7.48	*6.07*	0.82	5.41	*4.58*	*2.07*

Table 4. LogisticRegression

	StratifiedKFold			StratifiedGroupKFold			*GridSearch Difference*
Metric	Baseline	GridSearch	*Difference*	Baseline	GridSearch	*Difference*	
Balanced Accuracy	0.70	0.74	*0.03*	0.71	0.75	*0.04*	*0.01*
ROC-AUC	0.77	0.80	*0.04*	0.79	0.80	*0.01*	*0.00*
Precision	0.71	0.71	*-0.01*	0.71	0.73	*0.02*	*0.02*
Recall	0.62	0.75	*0.13*	0.64	0.73	*0.09*	*-0.02*
F1 Score	0.66	0.73	*0.07*	0.67	0.73	*0.06*	*0.00*
Fit Time (s)	1.19	6.76	*5.57*	1.33	4.40	*3.07*	*-2.36*

Table 5. RandomForest

	StratifiedKFold			StratifiedGroupKFold			*GridSearch Difference*
Metric	Baseline	GridSearch	*Difference*	Baseline	GridSearch	*Difference*	
Balanced Accuracy	0.73	0.70	*-0.03*	0.72	0.75	*0.03*	*-0.05*
ROC-AUC	0.81	0.83	*0.02*	0.79	0.83	*0.04*	*0.00*
Precision	0.74	0.71	*-0.02*	0.71	0.76	*0.05*	*-0.04*
Recall	0.67	0.63	*-0.05*	0.72	0.71	*-0.01*	*-0.08*
F1 Score	0.70	0.67	*-0.03*	0.71	0.73	*0.02*	*-0.06*
Fit Time (s)	1.62	8.21	*6.59*	2.24	6.87	*4.64*	*1.33*

Table 6. XGBoost

	StratifiedKFold			StratifiedGroupKFold			*GridSearch Difference*
Metric	Baseline	GridSearch	*Difference*	Baseline	GridSearch	*Difference*	
Balanced Accuracy	0.69	0.72	*0.03*	0.61	0.69	*0.07*	*0.04*
ROC-AUC	0.77	0.78	*0.02*	0.64	0.75	*0.10*	*0.04*
Precision	0.74	0.73	*-0.01*	0.62	0.67	*0.06*	*0.05*
Recall	0.57	0.67	*0.10*	0.58	0.65	*0.06*	*0.02*
F1 Score	0.64	0.70	*0.06*	0.59	0.66	*0.07*	*0.04*
Fit Time (s)	1.42	6.72	*5.30*	1.78	5.04	*3.25*	*1.69*

Table 7. KNearestNeighbors

	StratifiedKFold			StratifiedGroupKFold			*GridSearch Difference*
Metric	Baseline	GridSearch	*Difference*	Baseline	GridSearch	*Difference*	
Balanced Accuracy	0.63	0.70	*0.07*	0.59	0.65	*0.06*	*0.00*
ROC-AUC	0.70	0.73	*0.03*	0.66	0.74	*0.08*	*0.01*
Precision	0.66	0.66	*0.00*	0.62	0.62	*-0.00*	*0.00*
Recall	0.50	0.73	*0.23*	0.51	0.67	*0.15*	*-0.06*
F1 Score	0.56	0.69	*0.13*	0.54	0.64	*0.10*	*-0.05*
Fit Time (s)	0.65	4.98	*4.32*	2.11	7.01	*4.90*	*2.03*

Table 8. SVC, kernel=rbf

VBM Dataset

VBM Results							
Model Family	Model	**StratifyKFold**			**StratifyGroupKFold**		
		bACC	ROC-AUC	Time	bACC	ROC-AUC	Time
Tree-based	Random Forest	0.63	0.69	33s	0.60	0.70	22s
Non-linear	KNN	0.65	0.7	32s	0.64	0.62	20s

Table 9. Baseline models results for the VBM dataset

	StratifiedKFold			StratifiedGroupKFold		
Metric	Baseline	GridSearch	Difference	Baseline	GridSearch	Difference
Balanced Accuracy	0.66	0.64	-0.02	0.69	0.60	-0.09
ROC-AUC	0.74	0.72	-0.02	0.76	0.68	-0.08
Precision	0.68	0.66	-0.02	0.70	0.61	-0.10
Recall	0.54	0.52	-0.02	0.58	0.48	-0.10
F1 Score	0.60	0.58	-0.02	0.64	0.53	-0.10
Fit Time (s)	5.51	98.77	93.26	5.08	63.75	58.67

Table 10. RandomForest - VBM

3 Discussion

Index	ROIs Features (284 columns)				VBM Features (variable number of columns)					
	0	1	2	…	283	284	285	286	…	(n_columns - 1)
0	…	…	…	…	…	…	…	…	…	…
1	…	…	…	…	…	…	…	…	…	…
2	…	…	…	…	…	…	…	…	…	…
⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋱	⋮
n-1	…	…	…	…	…	…	…	…	…	…

Table 11. Structure of X_train data

References

1. Olabi, B., Ellison-Wright, I., Bullmore, E., Lawrie, S. M.: Structural brain changes in first episode Schizophrenia compared with Fronto-Temporal Lobar Degeneration: a meta-analysis. BMC Psychiatry **12**, 1-13 (2012). https://doi.org/10.1186/1471-244X-12-104/FIGURES/4. https://bmcpsychiatry.biomedcentral.com/articles/10.1186/1471-244X-12-104
2. DeLisi, L. E., Szulc, K. U., Bertisch, H. C., Majcher, M., Brown, K.: Understanding structural brain changes in schizophrenia. Dialogues in Clinical Neuroscience **8**(1), 71 (2006). https://doi.org/10.31887/DCNS.2006.8.1/LDELISI. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3181763/
3. Zipursky, R. B., Reilly, T. J., Murray, R. M.: The Myth of Schizophrenia as a Progressive Brain Disease. Schizophrenia Bulletin **39**(6), 1363-1372 (2013). https://doi.org/10.1093/SCHBUL/SBS135. https://dx.doi.org/10.1093/schbul/sbs135
4. Ahmed, M., Cannon, D. M., Scanlon, C., Holleran, L., Schmidt, H., McFarland, J., Langan, C., McCarthy, P., Barker, G. J., Hallahan, B., McDonald, C.: Progressive Brain Atrophy and Cortical Thinning in Schizophrenia after Commencing Clozapine Treatment. Neuropsychopharmacology **40**(10), 2409-2417 (2015). https://doi.org/10.1038/npp.2015.90. https://www.nature.com/articles/npp201590
5. Bonilha, L., Molnar, C., Horner, M. D., Anderson, B., Forster, L., George, M. S., Nahas, Z.: Neurocognitive deficits and prefrontal cortical atrophy in patients with schizophrenia. Schizophrenia Research **101**(1-3), 142 (2008). https://doi.org/10.1016/J.SCHRES.2007.11.023. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2441896/