BrainHack School

Humai

BrainHack 2024

Detecting ADHD through fMRI signals using machine learning classification models

Team 1:

Covelli, Francisco

Rodriguez, Agustin

Sirne, Ezequiel

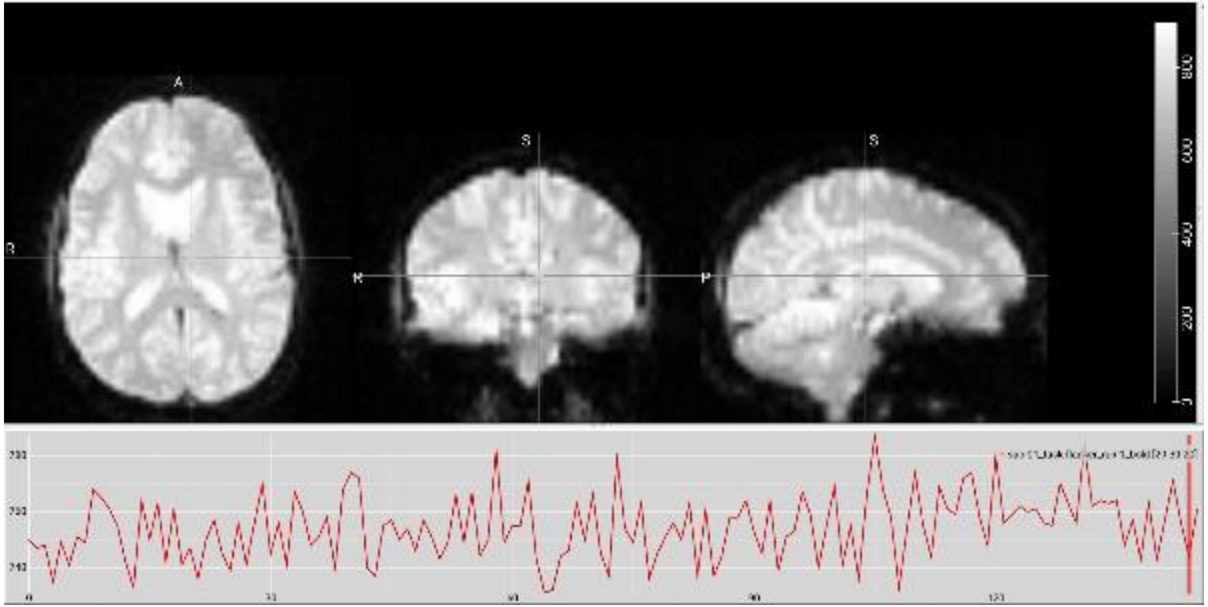
Project definition

- ADHD: inattention, impulsivity and hyperactivity
- 5% prevalence rate in Argentina
- Diagnosed with subjective clinical criteria that are hard to evaluate

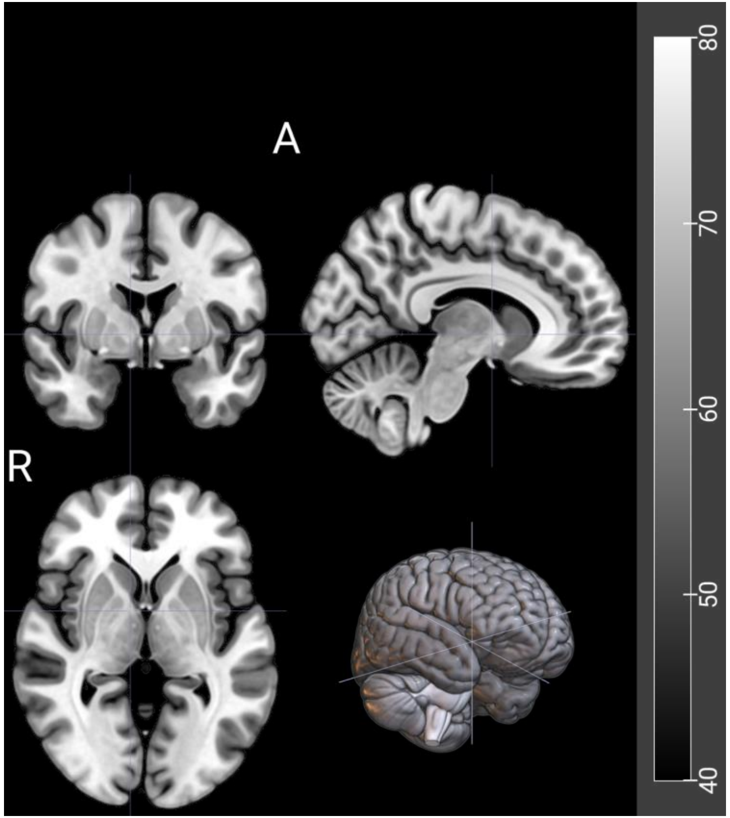
Our objective is to develop ML classification models to detect ADHD through the analysis of resting state functional MRI signals

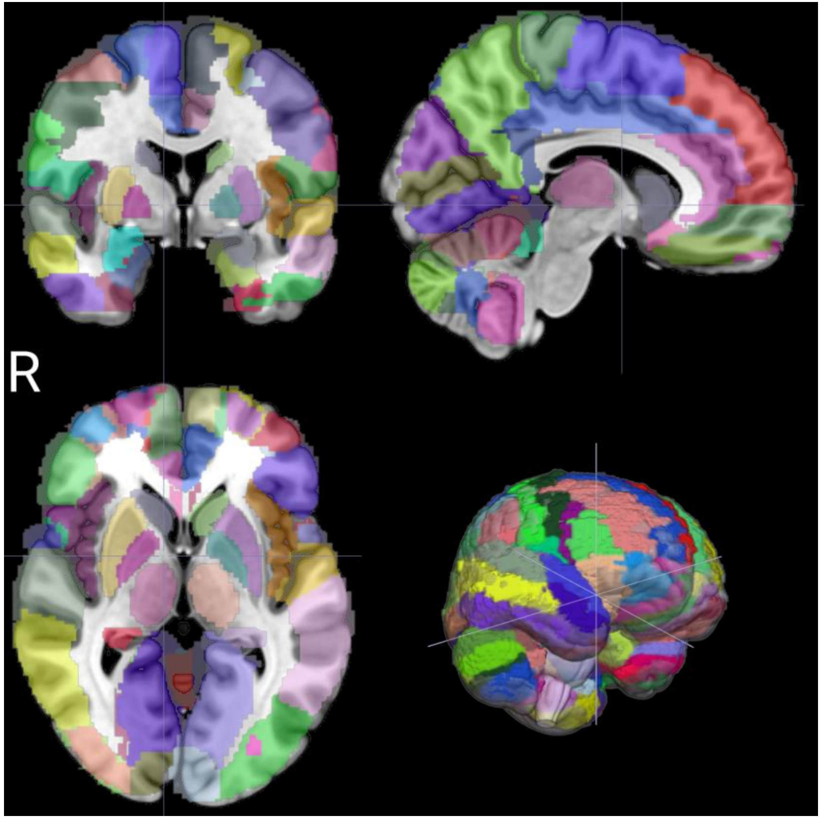
Functional MRI (fMRI)

- **BOLD Signal** (Blood Oxygenation Level Dependent)
- “*Neurovascular-coupling*”

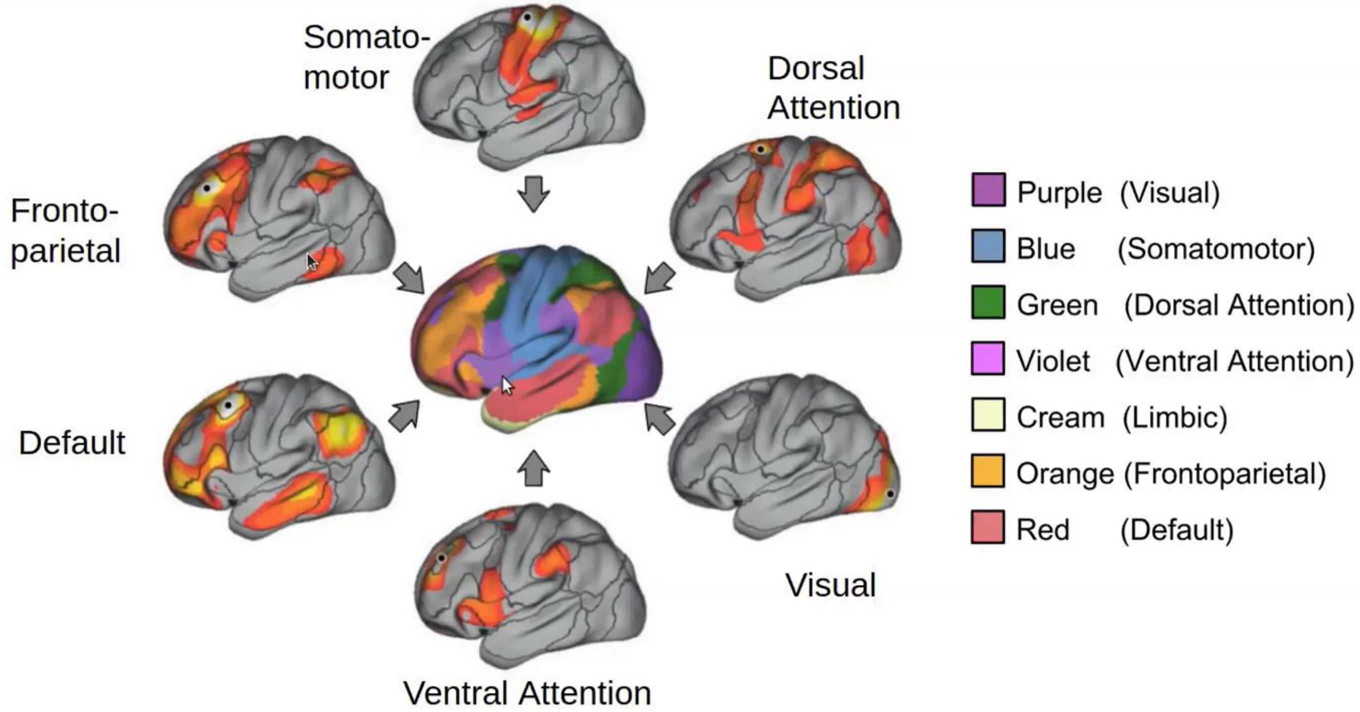


AAL Atlas Visualization with MRIcroGL





Yeo and Krienen's 7 clusters - Functional connectivity



Yeo BT, Krienen FM, Sepulcre J, Sabuncu MR, Lashkari D, Hollinshead M, Roffman JL, Smoller JW, Zöllei L, Polimeni JR, Fischl B, Liu H, Buckner RL. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. J Neurophysiol. 2011 Sep;106(3):1125-65. doi: 10.1152/jn.00338.2011. Epub 2011 Jun 8. PMID: 21653723; PMCID: PMC3174820.

Dataset Used

ADHD-200 Preprocessed Sample:

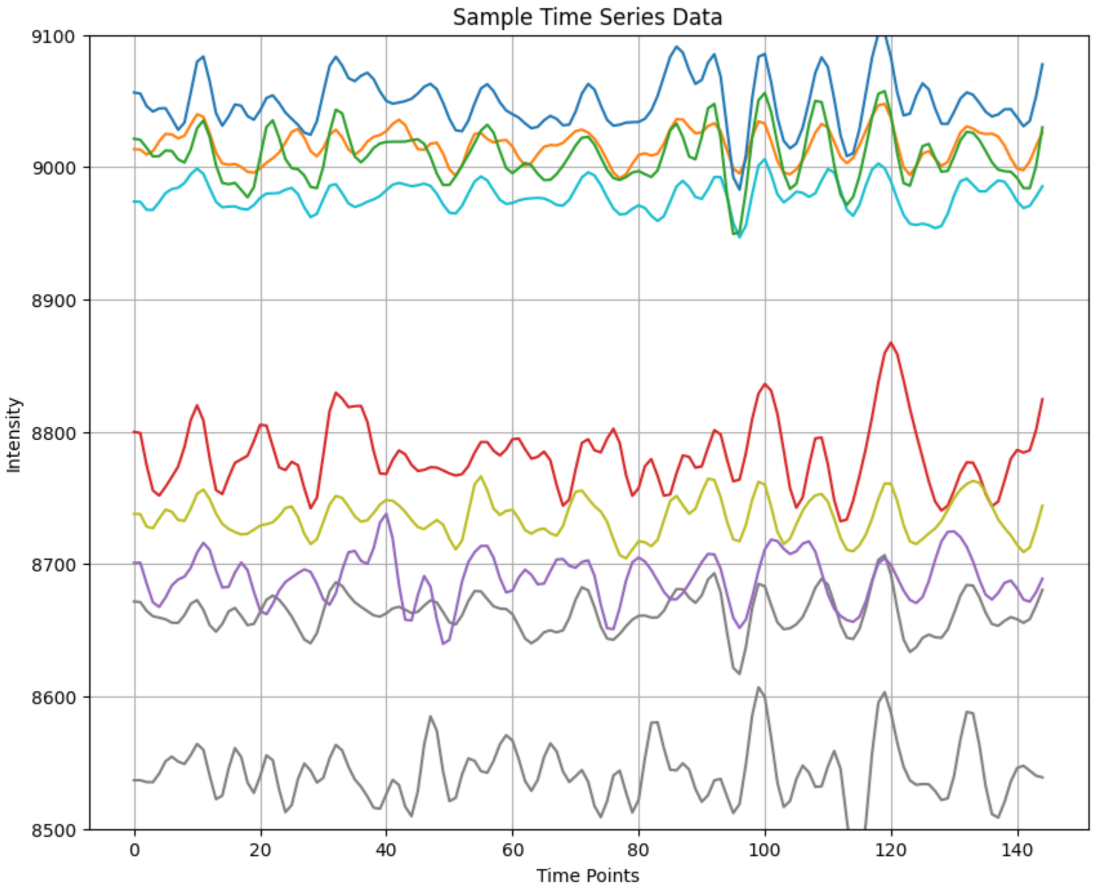
Prepared by the collaboration of 8 international neuroimaging institutes, who held a competition in 2011, to identify ADHD biomarkers.

Composition:

- 947 children and adolescents
- 362 diagnosed with ADHD
- 585 typically developing controls

In order to make the competition accessible to a broader range of researchers, the data was preprocessed and shared, what resulted in several publications, MS theses, PhD dissertations and even patents.

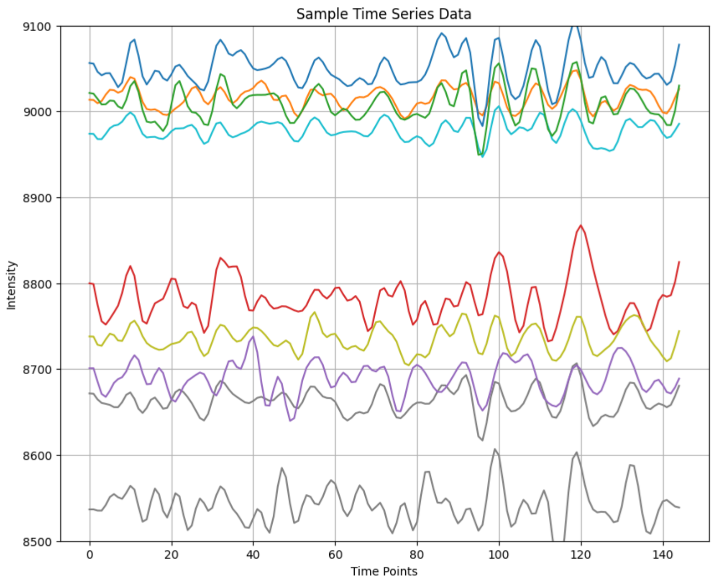
What we extracted from the Dataset:

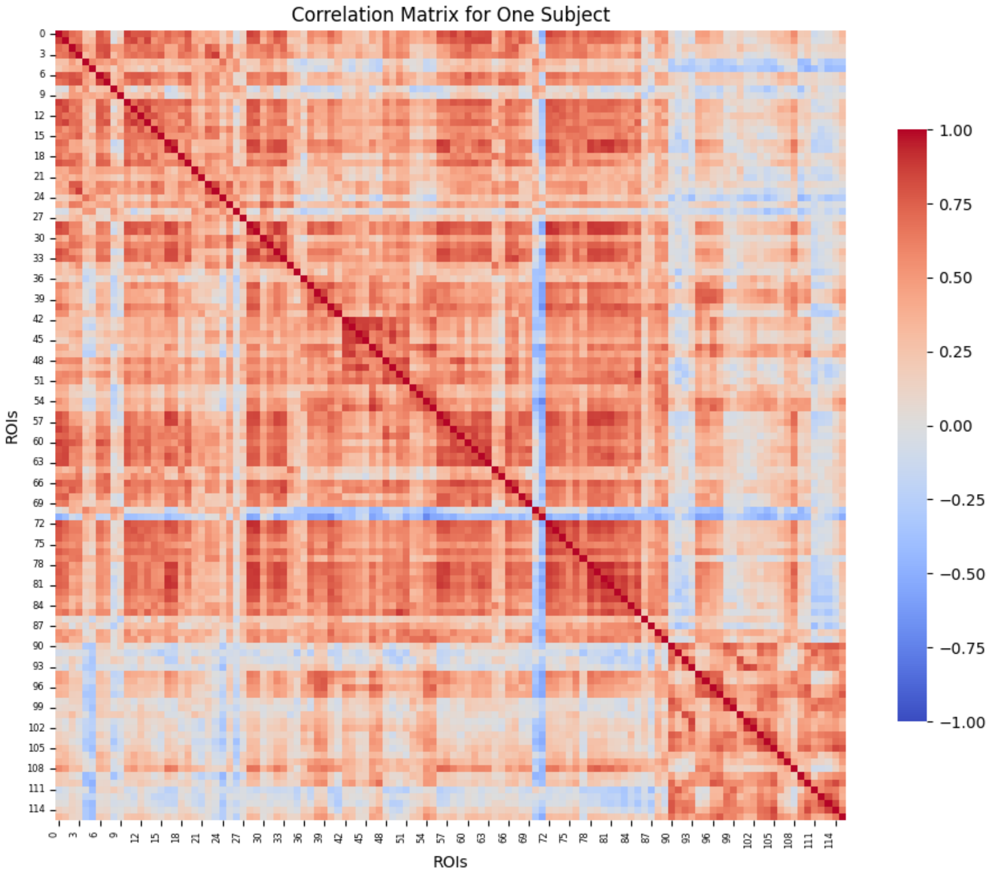


Each time series represents the intensity of the BOLD signal of every ROI (Region of Interest) along the rs-fRMI.

In order to create a connectivity matrix, we evaluated the Pearson Correlation between each of these signals.

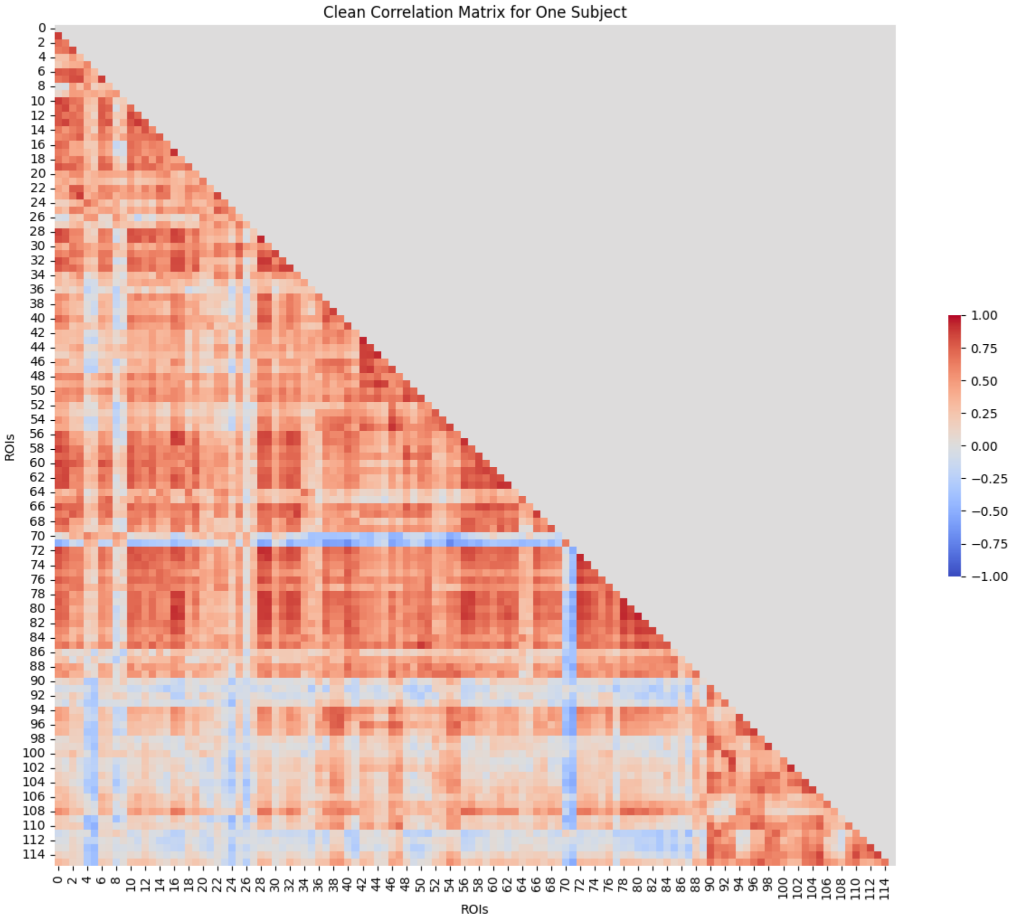
Connectivity or Correlation Matrix:





matplotlib seaborn

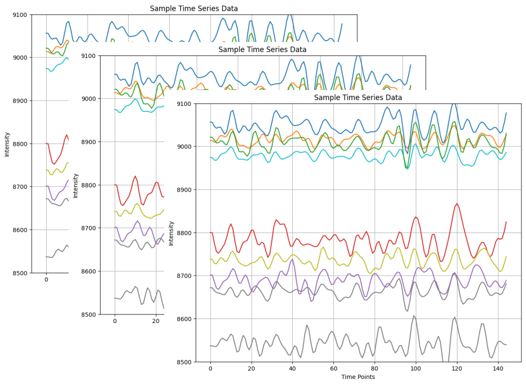
Clean Correlation Matrix and vectorizing:

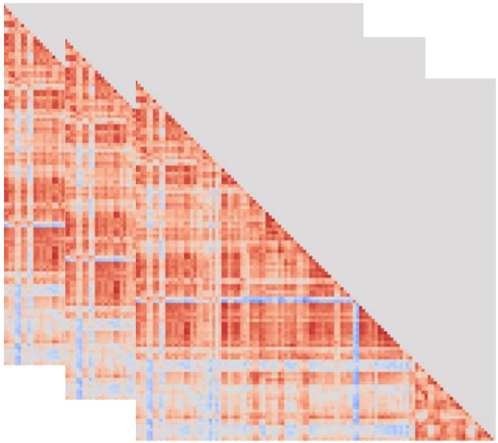


'Precentral_R x Precentral_L',
'Frontal_Sup_L x Precentral_L',
'Frontal_Sup_L x Precentral_R',
...
'Vermis_10 x Vermis_7',
'Vermis_10 x Vermis_8',
'Vermis_10 x Vermis_9'

This vector stores the Correlation Values of each ROI combination

Pipeline:





	0	1	2	3	4	5	6	7	8	9	...	6662	6663	6664	6665	6666	6667	6668	6669	subject	DX_GROUP
0	0.870813	0.805070	0.693176	0.778252	0.748457	0.844588	0.807095	0.737196	0.855809	0.769672	...	0.282437	-0.273149	0.228345	0.078729	0.017100	0.067529	-0.249568	-0.143400	50005	1
1	0.898271	0.530014	0.458091	0.251550	0.346688	0.300349	0.185360	0.055388	0.668472	0.039961	...	0.215894	0.395081	0.178194	0.047404	0.250613	0.364111	0.231605	0.175845	50006	1
2	0.694720	0.617756	0.640611	0.538903	0.731939	0.856895	0.400831	0.373279	0.515696	0.491006	...	0.663160	0.494288	0.250436	0.096148	-0.088067	0.275448	-0.038101	0.146421	50008	1
3	0.795809	0.704917	0.662008	0.580074	0.641586	0.697295	0.594052	0.528248	0.766270	0.575263	...	0.492290	0.206852	0.149637	0.104935	0.275394	0.167705	0.082875	0.731423	50009	1
4	0.698046	0.339043	0.658891	0.145301	0.412934	0.695916	0.194750	0.218813	0.408388	0.120781	...	0.121002	0.113757	0.132499	0.083765	0.192908	-0.120131	0.284368	0.443524	50010	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
815	0.608603	0.841734	0.548407	0.598860	0.691022	0.756287	0.512149	0.432697	0.604457	0.424698	...	0.439990	0.265809	-0.452533	0.323737	0.301574	0.438263	0.204875	0.497366	51574	1
816	0.781145	0.673924	0.553285	0.574009	0.709194	0.837303	0.436674	0.519944	0.494364	0.510137	...	0.406716	0.572137	-0.124384	-0.132317	0.308735	0.288378	0.383258	0.835765	51576	1
817	0.447159	0.478067	0.250900	0.076704	0.373822	0.664480	0.199131	0.068500	0.126183	0.174924	...	0.698613	0.489151	0.408260	0.340335	0.610982	0.521387	0.231358	0.410172	51581	1
818	0.423467	0.291398	0.000689	-0.106823	0.309935	0.306685	0.121247	0.081313	0.346079	0.270052	...	-0.108152	0.155436	-0.003550	0.074924	-0.230657	-0.037307	0.032230	0.110144	51606	1
819	0.830487	0.834947	0.623362	0.801337	0.773588	0.813550	0.723835	0.563768	0.864789	0.810816	...	0.340056	0.358228	0.332230	0.392646	0.129756	0.251530	0.119860	-0.004577	51607	1

Splitting the DataFrame:

Using train_test_split from sklearn we splitted the data with the following parameters:

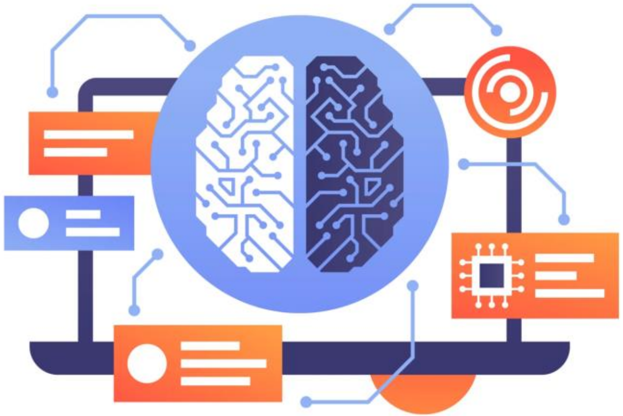
- test_size=0.2
- random_state=42
- stratify=target

Our Challenge:

Classification problem with a binary target using a Supervised Machine Learning Model.

In the first approach we explore the following models:

a. PCA (Principal Components Analysis)
b. LDA (Linear Discriminant Analysis)
c. Logistic Regression



Metrics:

In order to compare the models performances, we used:

- Train Accuracy
- Test Accuracy
- Confusion Matrix
- Classification Report, a tool from sklearn.metrics that includes several metrics (such as, precision, recall, f1-score and support)

XGBoost:

Since with the previous models we had low accuracy and high overfitting, we studied an ensemble model:

Extreme Gradient Boosting, combines the predictions of multiple individual models to produce a final prediction.

We selected this model hoping that its speed, performance, and versatility could help us reducing the overfitting and get better metrics.

Cross-Validation and Hyperparameters Optimization

Reading XGBoost documentation and using tools like GridSearchCV, Bayesian Optimization and RandomizedSearchCV, we tried several hyperparameters combinations in order to reduce overfitting.

We focused on RandomizedSearchCV since its lower processing cost:

```
param_dist = {
    'n_estimators': np.arange(50, 300, 50),
    'learning_rate': np.linspace(0.05, 0.2, 4),
    'max_depth': np.arange(2, 5, 1),
    'min_child_weight': np.arange(1, 5, 1),
    'subsample': np.linspace(0.6, 0.9, 3),
    'colsample_bytree': np.linspace(0.6, 0.9, 3),
    'alpha': np.linspace(0, 1, 3)
}
```

Complexity Control

Randomness

Regularization

Final Results:

- ❖ Train Accuracy: 0.9939
- ❖ Test Accuracy: 0.6219
- ❖ F1 Score: 0.5867
- ❖ ROC AUC: 0.6192

```
Confusion Matrix:
[[58 31]
 [31 44]]
Classification Report:
              precision    recall  f1-score   support

           0       0.65      0.65      0.65        89
           1       0.59      0.59      0.59        75

    accuracy                           0.62       164
   macro avg       0.62      0.62      0.62       164
weighted avg       0.62      0.62      0.62       164
```

Table 5. Published resting-state fMRI-based ADHD classification studies.

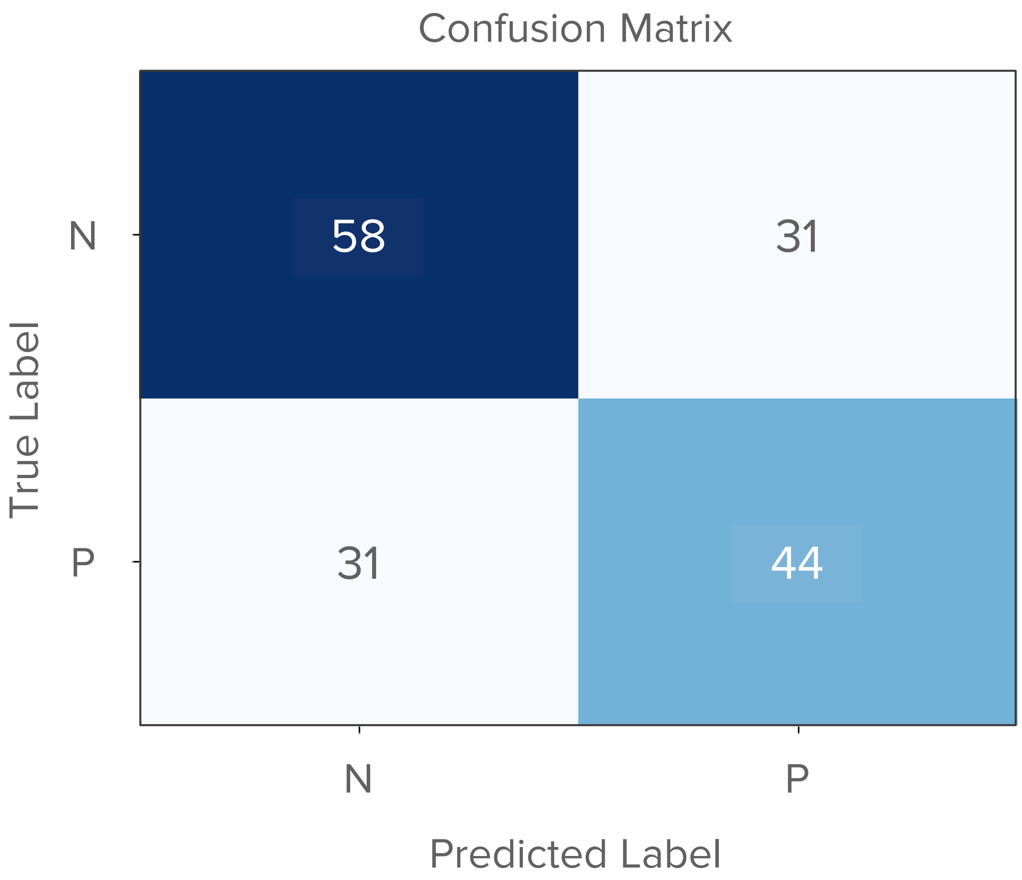
ACC	AUC	Number of subjects	Classifier	Dataset	References
0.6916	0.741	775	MKL	ABCD	**Wang et al.**
0.688	0.70	776	Sparse LR	ADHD-200	Zhan et al. [73]
0.7884	-	645	T-R-SVM	ADHD-200	Shao et al. [62]
0.6491	-	729	SVM	ADHD-200	Sen et al. [74]
0.6604	-	730	CNN	ADHD-200	Zou et al. [36]
0.597	-	940	SVM	ADHD-200	Ghiassian et al. [75]
0.54	-	929	LR	ADHD-200	Sato et al. [30]
0.6667	-	839	SVM	ADHD-200	Sidhu et al. [76]
0.6959	-	947	PCA-LDA	ADHD-200	Dey et al. [61]

Multiple measurement analysis of resting-state fMRI for ADHD classification in adolescent brain from the ABCD study - Wang et al, 2023

0.6219	0.619	820	XGBoost	ADHD-200	Ours

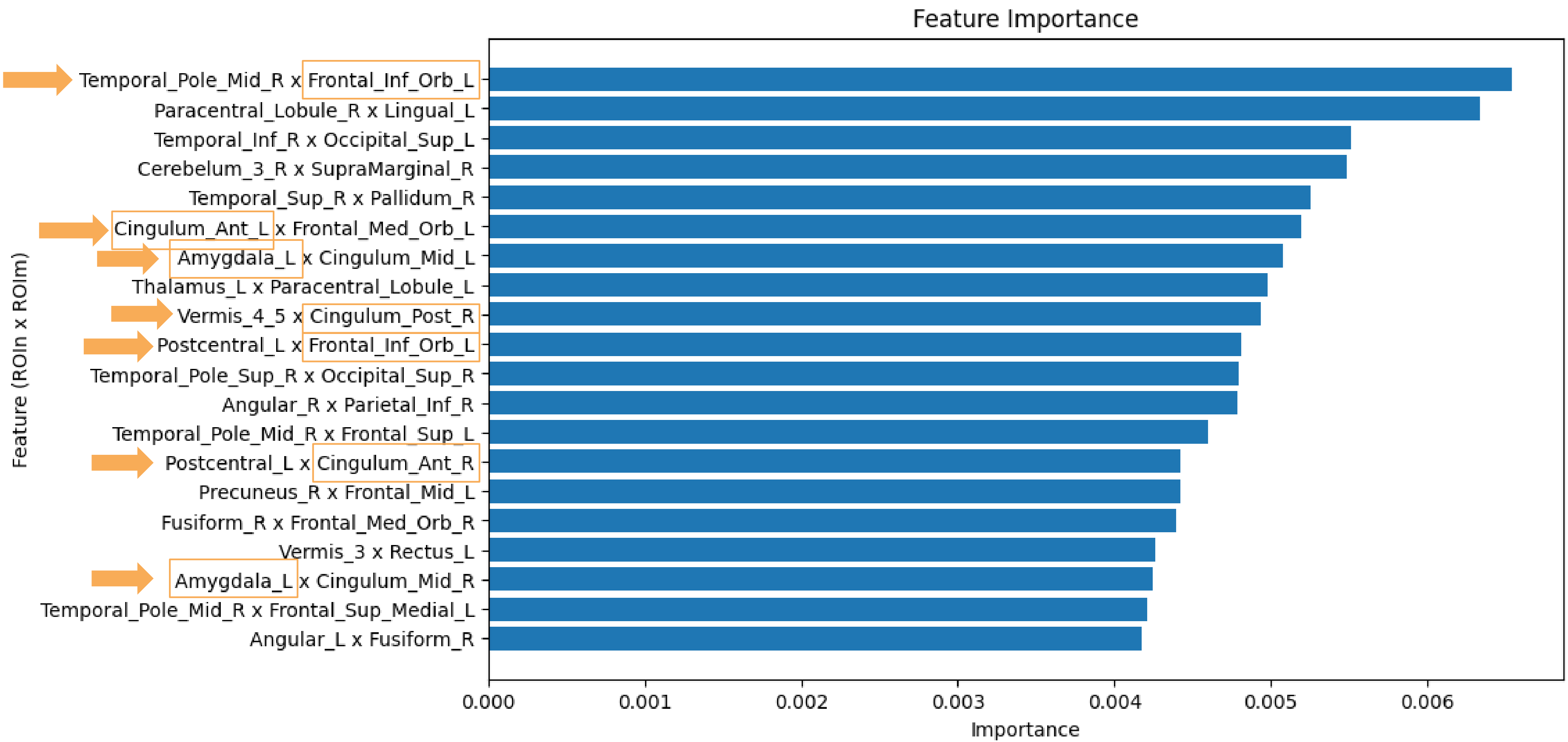
Covelli, Rodriguez, Sirne - Detecting ADHD through fMRI signals using machine learning classification models

Final Results:

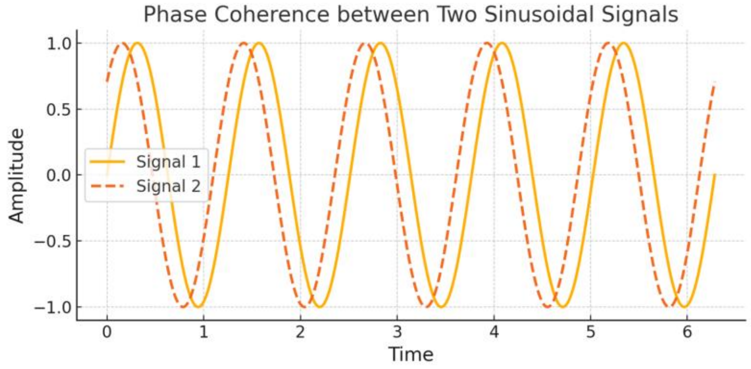


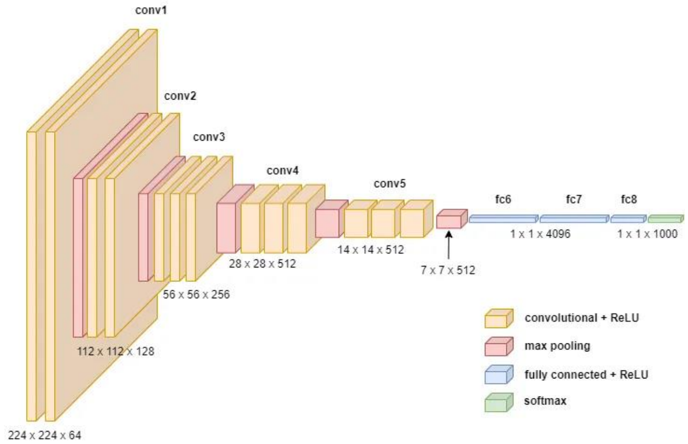
Despite having reduced both Type I and Type II errors, we can conclude that with an accuracy of 0.6219, there is significant room for improvement in ADHD diagnosis based on classification models.

Top 20 Feature Importance & Connectivity Networks ROIs:



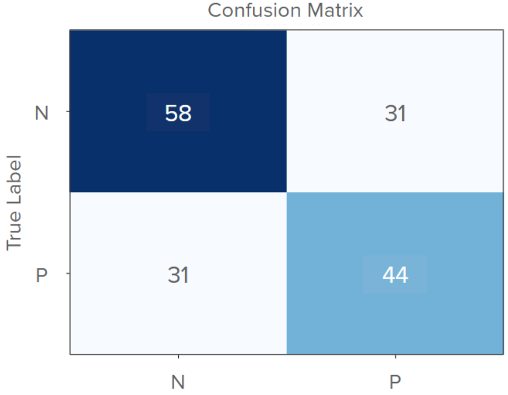
Other possible approaches:





Other features

More complex models



Sensitivity and Specificity

Covelli, Rodriguez, Sirne - Detecting ADHD through fMRI signals using machine learning classification models

Conclusion and acknowledgement

There are many ways to expand and improve this project, and we are excited to explore them.

Finally, we would like to express our gratitude to our **teaching assistants** from **Humai**, **Buenos Aires**, who have supported us from day one and generously shared their diverse expertise in neuroscience and computer science.

Feel free to contact us!

If you need assistance with your artificial intelligence project, we would be happy to help!

Francisco Covelli
fcovelli@med.unlp.edu.ar

Agustin Rodriguez
agurodri96@hotmail.com

Ezequiel Sirne
ezequielsirne@gmail.com

BrainHack School

Humai

BrainHack 2024

Thank you for your attention!

Team 1:

Covelli, Francisco

Rodriguez, Agustin

Sirne, Ezequiel