NOVA IMS Information Management School

Mestrado em Data Science and Advanced Analytics, com especialização em Data Science

Seizure Detection with EEG and Big Data

Project of Big Data Analytics

Abstract

This project explores the application of Big Data techniques to detect epileptic seizures in pediatric patients. Using real EEG recordings from the EEG database, we developed a complete data pipeline—from preprocessing raw signals to training and evaluating predictive models. The main goal was to classify brain activity into three states: normal, pre-seizure, and seizure. To achieve this, we implemented and compared four different machine learning models: Logistic Regression, Random Forest, Gradient Boosted Trees, and Multilayer Perceptron, using PySpark and Databricks. Random hyperparameter search and 3-fold cross-validation were employed to optimize each model. Among the evaluated models, Random Forest achieved the best overall performance, especially in handling imbalanced data and partially identifying seizure activity. This project demonstrates how Big Data analytics can be leveraged to develop intelligent tools that support healthcare decision-making and improve the lives of patients with epilepsy.

Group N

David Cascão, 20240851
Jan-Louis Schneider, 20240506
Jorge Cordeiro, 20240594
Marta Boavida, 20240519
Sofia Gomes, 20240848

Spring Semester 2024-2025

Table of Contents

1 Introduction & Background

In this project, we decided to focus on a topic that combines two areas with a profound impact on people's lives: healthcare and data science. More specifically, we explored how Big Data can be used to help detect epileptic seizures in children, using real EEG recordings collected at Children's Hospital Boston.

These recordings come from pediatric patients with severe epilepsy who were closely monitored over several days. During this time, their anti-seizure medication was carefully reduced, allowing doctors to observe seizure patterns more clearly and assess whether the child might benefit from surgical treatment.

Epileptic seizures are sudden bursts of abnormal electrical activity in the brain. For people with epilepsy, these episodes can happen without warning, often putting them at risk of injury or worse. Being able to detect a seizure as it begins can be life-changing. It could mean alerting a caregiver in time to help, or even automatically triggering a response from a medical device.

This is where Big Data comes in. EEG data is complex, high-volume, and continuous-making it a perfect candidate for advanced data analytics and machine learning. By processing large amounts of brainwave data, we aim to identify patterns that can help predict or detect seizures more effectively.

Our motivation for this project is both technical and human. We want to show how data-driven approaches can lead to smarter healthcare tools, and hopefully, contribute to improving the quality of life for patients living with epilepsy.

Through this report, you can get a detailed explanation of everything that was done to achieve our goal. The code that made this project possible is available at the following link `https://github.com/Gomsofi06/BigDataAnalysis`.

2 Data Collection & Preprocessing

We used a public dataset available through PhysioNet [6], a well-known platform for sharing physiological data for research purposes.

The dataset consists of unstructured time-series data. Each file contains EEG signals recorded from multiple electrodes placed on the patient's scalp, with sampling rates high enough to capture detailed electrical activity in the brain. These recordings span hours or even days, and include both seizure and non-seizure events. In addition to the raw EEG data, metadata is provided to mark the start and end times of each seizure, which is essential for labeling and training supervised learning models.

Working with this type of biomedical data presents unique challenges: the signals are complex, often noisy, and vary significantly between patients. So we went through several cleaning and pre-processing steps. First we did an exploration of the data in order to see the data types, the missing values, check for duplicates and then, we start to preprocessing this data: deal with incoherences, handle missing values and remove outliers.

Once we imported the necessary libraries and loaded the dataset, we began by splitting the data into three subsets: 70% for training, 15% for testing, and 15% for validation. We made sure that each split preserved the proportion of the different labels, so that our models would be trained and evaluated on balanced data.

To better understand the structure and content of the dataset,we began by exploring it in detail. The dataset contains 4,608,000 rows and 26 columns, with each column representing a specific EEG signal channel or metadata. We looked at the first few rows and reviewed all columns along with their data types. During this step, we noticed that two specific EEG signal channels, FP1-F7 and FT10-T8, had a string data type, although they should be double. A closer look revealed that the only non-numerical values in those columns were 'inf', likely introduced by the Spark processing system when handling corrupted or unreadable values. Fortunately, these cases were rare, affecting only about 0.002% and 0.001% of the rows, respectively. Since we couldn't determine the original values, we decided to treat these entries as missing values.

Next, we checked for duplicated rows, and none were found. We then calculated the percentage of rows with missing values, which turned out to be just 0.26%. Given this small proportion, and considering that EEG data tends to come in continuous blocks of similar values (especially within the same class), we decided it was reasonable to drop those rows.

Outlier detection was also an important part of our preprocessing. Many of the EEG columns showed extreme values like ±1,000,000, which we suspect resulted from equipment malfunction during data recording. Here we have an small example of the variables FP1-F7 and FT10-T8 [1]. To avoid these distortions, we excluded those outliers from our analysis. Additionally, we inspected the next 20 highest and lowest values (excluding the ±1,000,000s) and determined that a reasonable threshold for the signal range would be between -0.1 and 0.1. Any values outside this range were considered outliers and removed. In total, about 2.3% of the rows were dropped. Here is an excerpt from the FP1-F7 and FT10-T8 variables after removing the outliers [2].

To gain deeper insight into the EEG patterns around seizures, we created visualizations focusing on the borders of seizure events. We used the following label classification throughout the project, Label 0: normal brain activity, Label 1: pre-seizure state, Label 2: during a seizure. Then, we looked at data segments just

before the beginning [3] and at the end [4] of the first seizure block (label 2). When analyzing these visualizations, we observed that the EEG signal patterns for labels 0 (normal state) and 1 (pre-seizure state) are extremely similar in amplitude and frequency. This makes it very difficult for machine learning models to distinguish between these two classes based solely on signal values. In contrast, the seizure state (label 2) shows a clear and distinct change in the signal, which is easier for the models to learn. Moreover, since the dataset is heavily imbalanced—with label 0 dominating the majority of the data—the models tend to favor predicting label 0, and almost completely ignore label 1. This is not necessarily a failure of the model or pipeline, but rather a reflection of the data distribution. It is likely that if the task were a binary classification between label 0 and label 2, the results would be significantly better in terms of precision and recall.

3 Methodology & Tools

To approach the task of seizure detection, we developed a consistent modeling pipeline applied across four different machine learning models: Logistic Regression[1], Random Forest[3], Gradient Boosted Trees[4], and Multilayer Perceptron[2]. All models were implemented using PySpark, with random search for hyperparameter tuning and 3-fold cross-validation to ensure fair evaluation.

We ran all experiments using Databricks [5]. Our pipeline followed the same structure for each model: first, we did data preparation using VectorAssembler to convert EEG features into a format compatible with PySpark's ML algorithms, then we used a random hyperparameter sampling, rather than exhaustive grid search, to find effective model configurations efficiently. In order to evaluate candidate models and identify the best set of parameters we used Cross-validation and finally, we used metrics such as accuracy, precision, recall, F1-score, confusion matrix, and ROC/AUC to evaluate the validation and test sets.

This unified approach ensured consistency across models, making comparisons more meaningful. Despite sharing the same workflow, each model introduced its own characteristics:

- Logistic Regression: Served as a strong, interpretable baseline with relatively simple decision boundaries. With this model we obtained the following results [5]. We can conclude that this model performs very well on class 0, with high precision (0.87) and recall (0.99). However, it completely fails to detect classes 1 and 2, with near-zero f1-scores. This suggests that the model is strongly biased toward the majority class and struggles with class imbalance.

- Random Forest: Used ensemble learning to capture non-linear patterns through multiple decision trees. With this model we obtained the following results [6]. After an analysis, we can conclude that Random Forest shows excellent performance on class 0 and some ability to detect class 2 (f1-score 0.37). However, it fails entirely on class 1 (f1-score 0.00). It has the highest overall accuracy

(0.92) and the best balance between precision and recall among all models, making it the most robust of the three—though still weak on minority classes.

- Gradient Boosted Trees: Built trees sequentially to correct errors of previous ones, often achieving better performance at the cost of higher training time. This model didn't run to completion due to databricks'time constraints.

- Multilayer Perceptron (MLP): Modeled complex, non-linear relationships using a neural network with tunable hidden layers and iterations. With this model we obtained the following results [7]. The MLP model, like the others, performs well only on class 0 (f1-score 0.95) and fails completely on classes 1 and 2. Although it achieves high accuracy (0.90), it lacks the ability to handle minority classes, limiting its usefulness for imbalanced classification tasks.

We also considered implementing XGBoost, a powerful gradient boosting method, but were limited by the Databricks Community Edition, which doesn't support external library installations. For this reason, XGBoost was not included in our final evaluation.

Additionally, we experimented with strategies to address class imbalance, such as adjusting weights and tuning hyperparameters accordingly. However, these attempts did not lead to significant performance improvements, particularly for the minority classes.

Also, although our initial goal included experimenting with streaming data, the structure of the EEG dataset (which is entirely static and file-based) did not support this approach. True streaming would require real-time, continuous input - something not available in this dataset.

4 Comparison of models

To better compare and interpret the performance of our models, we also used Tableau (with the support of GraphFrames)[7] to visualize key evaluation metrics such as Precision, Recall, and F1 Score across the different models and classes.

Although the interactive version of this dashboard is available on our GitHub repository, we included a static image in the report to provide a clear overview of the results and support our analysis.

From the visual comparison [8], the Random Forest classifier proved to be the most efficient overall. It achieved the highest test accuracy (92%) and outperformed the other models in terms of macro and weighted F1-scores. While all models struggled with detecting minority classes, Random Forest was the only one that managed to partially identify class 2, resulting in a significantly better F1-score for that class. This suggests that Random Forest offers a more balanced and robust performance, especially in scenarios involving class imbalance.

Using visualization tools like Tableau helped us not only validate numerical results, but also better understand how each model behaves across different classes. This was particularly useful in identifying trade-offs between metrics and ensuring we chose the most reliable and balanced model for detecting seizures.

5 Conclusion

Throughout this project, we built a scalable and consistent big data pipeline capable of handling complex EEG data for seizure detection. After evaluating four different models, we found that while Logistic Regression and MLP performed well on the majority class, they failed to detect minority classes, such as seizures. Gradient Boosted Trees could not be fully evaluated due to platform limitations. Random Forest stood out as the most efficient model, achieving the highest accuracy and offering a more balanced classification across different brain states. Although performance on minority classes remains a challenge, our results highlight the potential of ensemble models in healthcare analytics.

Future work could explore advanced techniques like oversampling, cost-sensitive learning, or real-time streaming solutions to improve classification performance and enable earlier seizure detection. Additionally, applying the pipeline to EEG data from multiple patients - rather than focusing on a single subject—could allow the development of a more generalized model, capable of adapting to different brain signal patterns and enhancing its clinical applicability.

References

[1] Apache Spark. *PySpark ML Documentation: LogisticRegression*. Accessed: 2025-05-14. 2025. URL: https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.LogisticRegression.html.

[2] Apache Spark. *PySpark ML Documentation: MultilayerPerceptronClassifier*. Accessed: 2025-05-14. 2025. URL: https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.MultilayerPerceptronClassifi
html.

[3] Apache Spark. *PySpark ML Documentation: RandomForestClassifier*. Accessed: 2025-05-14. 2025. URL: https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.RandomForestClassifier.html.

[4] Apache Spark. *PySpark MLlib Documentation: GradientBoostedTrees*. Accessed: 2025-05-14. 2025. URL: https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.mllib.tree.GradientBoostedTrees.html.

[5] Databricks. *Databricks - Data Analytics and AI Platform*. Accessed: 2025-05-14. 2025. URL: https://www.databricks.com.

[6] Ary L. Goldberger et al. *CHB-MIT Scalp EEG Database*. Accessed: 2025-05-14. 2024. URL: https://physionet.org/content/chbmit/1.0.0/chb01/.

[7] Tableau Software. *Tableau - Visual Analytics Platform*. Accessed: 2025-05-14. 2025. URL: https://www.tableau.com.

A Appendix

Figure 1: Outliers

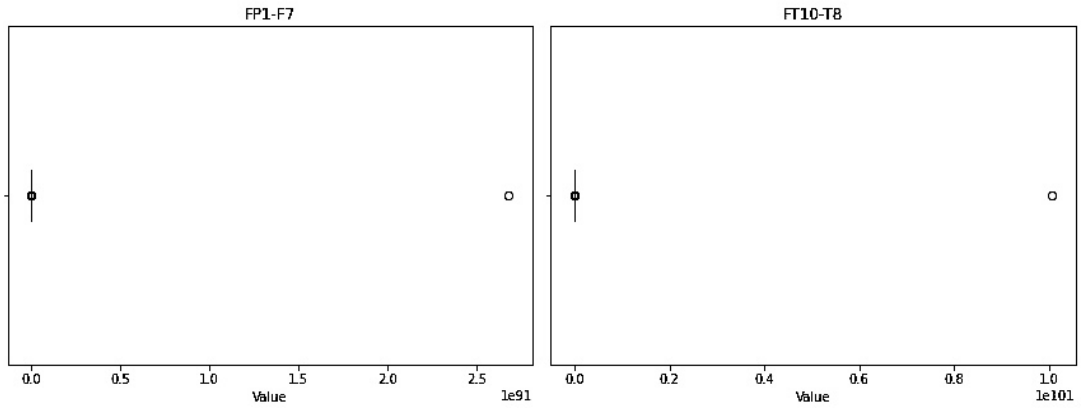

Figure 2: Outliers treated

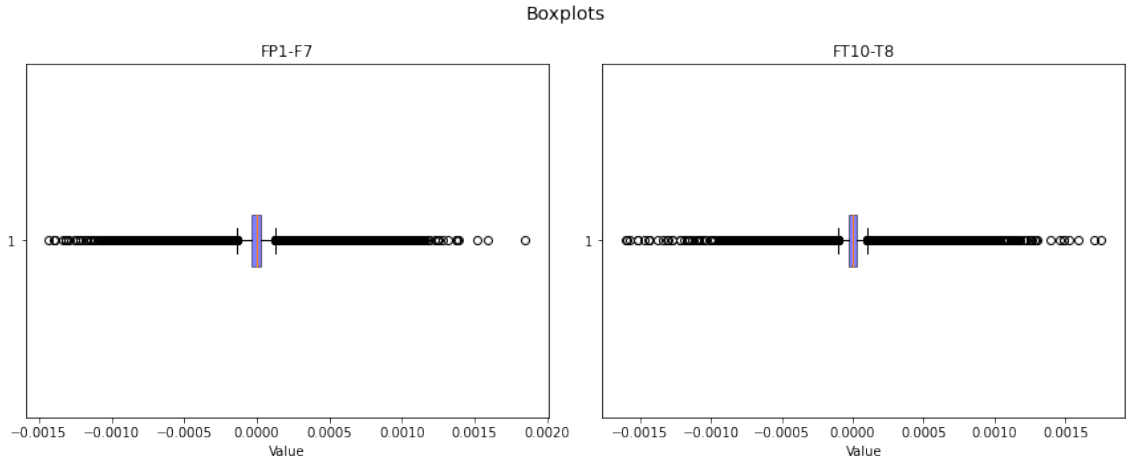

Figure 3: Area around beginning of first block of data with label 2

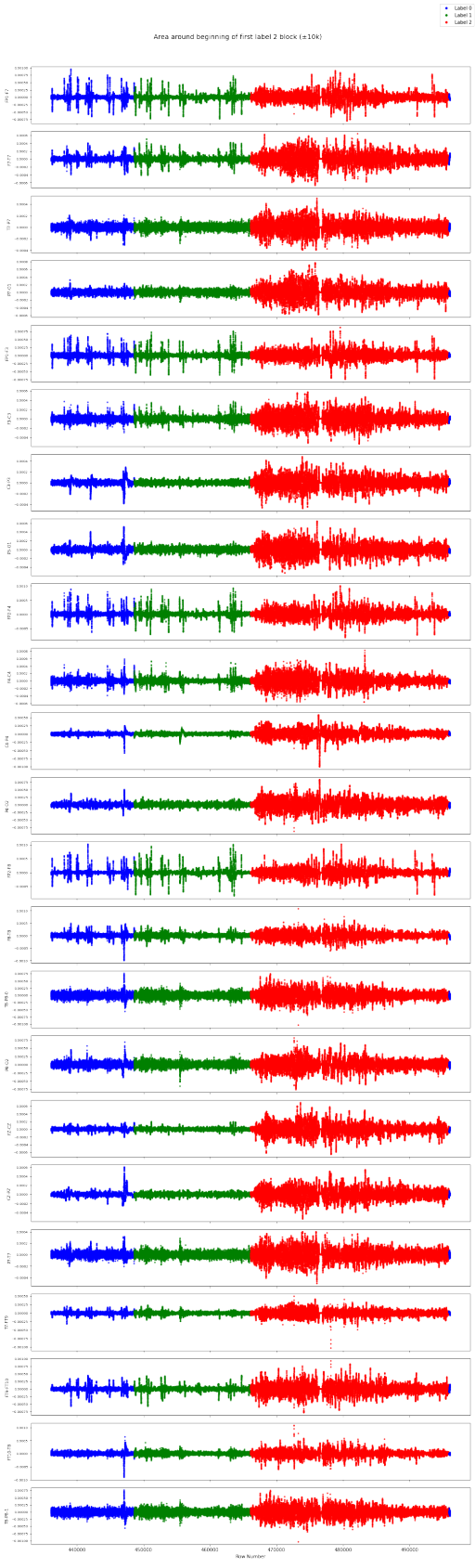

Figure 4: Area around end of first block of data with label 2

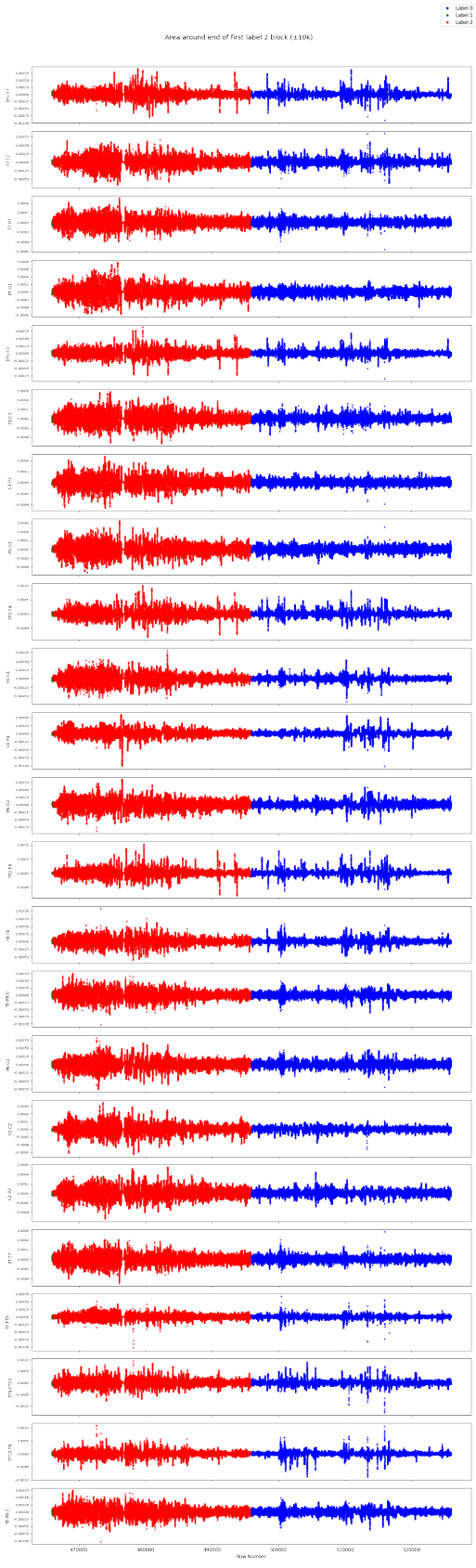

Figure 5: Evaluation of Logistic Regression Model

```
Logistic Regression (Test) Classification Report:
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.99      0.92    589564
           1       0.03      0.01      0.01     25185
           2       0.09      0.00      0.00     66646

    accuracy                           0.86    681395
   macro avg       0.33      0.33      0.31    681395
weighted avg       0.76      0.86      0.80    681395
```

Figure 6: Evaluation of Random Forest Model

```
Classification Report:
              precision    recall  f1-score   support

           0       0.92      1.00      0.96    615738
           1       0.00      0.00      0.00     25266
           2       0.93      0.23      0.37     40377

    accuracy                           0.92    681381
   macro avg       0.62      0.41      0.44    681381
weighted avg       0.88      0.92      0.89    681381
```

Figure 7: Evaluation of Multilayer Perceptron Model

```
Classification Report:
              precision    recall  f1-score   support

           0       0.90      1.00      0.95    615738
           1       0.00      0.00      0.00     25266
           2       0.00      0.00      0.00     40377

    accuracy                           0.90    681381
   macro avg       0.30      0.33      0.32    681381
weighted avg       0.82      0.90      0.86    681381
```

Figure 8: Visualization of comparision metrics with Tableaux