REGRESSION METHODS FOR CUTTING-EDGE PROBLEMS

ABSTRACT

Regression methods play an important role in solving cutting-edge problems, including those in neuroscience, computer vision, and natural language processing. Although cutting-edge problems often involve large, complex datasets, regression methods can process this data into useful information. From modeling relationships between different features to predicting outcomes using supervised learning, real-life challenges can be tackled effectively. There is a wide variety of regression methods, such as linear, non-linear, and tree-based approaches. It is indispensable to recognize that each regression method has distinct features for addressing different kinds of challenges. For instance, linear regression methods can uncover hidden patterns in data and predict outcomes for unknown data. In this paper, decision tree regression, a tree-based regression method, will be employed to analyze a complex dataset consisting of brain, image, and text features. Regarding scalability, interpretability, and feasibility, regression provides a sensible approach to analyzing cutting-edge problems and uncovering hidden insights within complex datasets.

Introduction

In modern society, everything is digitalized. Challenges have been steadily increasing due to the surge of complex, high-dimensional datasets. It is undeniable that simplifying datasets has become a crucial task for addressing cutting-edge problems. Analyzing and training features can provide deeper insight into datasets and enable effective modeling of the relationships between different features.

Regression methods have always been a key tool for analyzing complex datasets, as they reveal significant relationships. Various regression techniques are available for machine learning and analysis, including linear regression, logistic regression, ridge regression, lasso regression, etc. However, decision tree regression is selected for this analysis due to its outstanding performance in handling non-linear data with complex distributions.

Related Work

In this paper, we explore a multimodal AI dataset as an example of machine learning. The EEG BraVL dataset, comprising brain, image, text, and label features, is employed for data exploration, implementation, model training, and further analysis. As emphasized in the abstract, a regression method, specifically decision tree regression, has emerged as a powerful tool to explore the EEG dataset. This tree-based regression method processes non-linear data by hierarchically splitting it into leaves based on distinct features. Ultimately, label predictions are achieved by training the data within the tree, demonstrating the effectiveness in simplifying complex datasets.

Supervised learning will be applied to the provided dataset. It is crucial to note that the configuration of decision tree regression can yield diverse results, including the number of splits, the size of leaf nodes, the number of depth searches, etc. This hierarchical tree structure aims to create a model that predicts the target variable based on the distribution of child nodes. As data is allocated into various nodes, a tree structure is constructed. Subsequently, the implementation's performance will be measured and refined, new methodology will also be discussed based on the test results. Notably, the accuracy and performance of the model will be significantly improved.

1 Project

This section will cover four sub-sections: Comprehensive Dataset Exploration, Custom Model Implementation, Result Analysis and Visualisation, Paradigm Design, and Data Splitting.

1.1 Comprehensive Dataset Exploration

1.1.1 KNOW

In this section, we aim to gain a deeper understanding of the dataset through basic statistical analysis. The EEG BraVL dataset is utilized for the implementation of machine learning. This dataset has 19417960 data items in total, with 1654 classes in the seen label category and 200 classes in the unseen label category. The brain feature data consists of 561 features, each with 16540 data. Similarly, the image feature data contains 100 features with 16540 data each and the text feature data contains 512 features with 16540 data each. The label feature data has one column with 16540 data points. The brain, image, and text data will be used in conjunction with the label data for supervised learning purposes. However, the class of the dataset will be reduced to 20 for enhancing high efficiency with easier implementation.

Given the dataset's size, exploration is critical before processing data. To uncover hidden patterns, all data are iterated to calculate statistical measures such as count, mean, std, min, 25%, 50%, 75%, max. Following this, we identify the most frequently occurring range of values based on the data range of each data frame. It is found that the most frequent brain data range is 0.0461 to 0.3733, for the image data is -3.5633 to 14.7623, and for text data is -0.6018 to -0.1584. Additionally, the peak amplitude of the brain data, representing the strength and intensity of brain signals, is identified as 9.1154, highlighting its significance in the analysis.

1.1.2 SEE

To visualize the dataset, we will use box plots, violin plots, and heatmaps for each of the features.

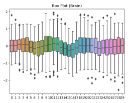

Figure 1: brain (box)

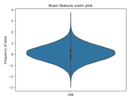

Figure 2: brain (violin)

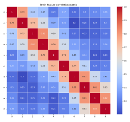

Figure 3: brain (heatmap)

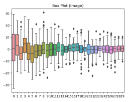

Figure 4: image (box)

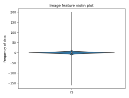

Figure 5: image

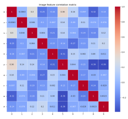

Figure 6: image(heatmap)

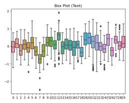

Figure 7: text (box)

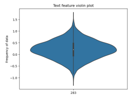

Figure 8: text (violin)

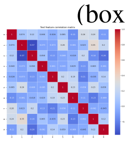

Figure 9: text (box)

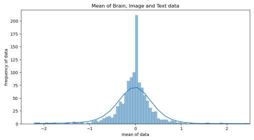

Figure 10: mean of all data (histogram)

1.1.3 FIND

The box plot illustrates the distribution of data for 30 randomly selected features from the dataset, it reveals high consistency in the features of the brain dataset, whereas the image dataset exhibits a diverse data range, and the text dataset shows a fluctuating data range. Data cleaning is necessary for both the image and the text data to achieve an optimal result.

The violin plot highlights that the distribution of the image features differs significantly from the other two feature types, exhibiting low variability with similar data points. This could suggest either a highly concentrated dataset or insufficient data for accurately representing a normal distribution. However, the latter possibility is unlikely, given that we have 100 features with 16,540 data points for each.

The heatmap emphasizes the stark difference between the brain features, with image and text features. The gradient of color in brain features indicates that data is distributed consistently, while image and text features appear less organized. This consistency makes the brain features ideal for testing the performance of the regression method.

The histogram reveals the presence of an outlier in the range of 0.1 to 0.2, where the frequency of data points significantly deviates from the expected pattern, exceeding the typical bell curve shape of a normal distribution. This observation suggests that the data does not adhere to a normal distribution, leading to inaccurate performance.

1.2 Custom Model Implementation

1.2.1 IMPLEMENT

In the era of data-driven decision-making, identifying meaningful patterns from complex, multi-dimensional datasets is essential. **Decision Tree Regression** can be one of the most powerful tools for implementing the machine learning process with ease. This tree-based regression method recursively splits the dataset into smaller subsets, capturing intricate relationships between inputs and outputs. EEG dataset consists of high-dimensional data, and based on the results obtained in FIND, it can be observed that the dataset contains substantial noise, making it hard to analyze. Therefore, PCA, a dimensionality reduction technique, has been first implemented on brain data. As the sample data for machine learning, it has been a better practice to test the algorithm with a more organized subset. Specifically, 20 seen classes, along with 50 selected features that contain 200 data items will be utilized for testing the algorithm. It is important to note that this process is equally applicable to the other two datasets.

Several hyperparameter configurations have been implemented. The maximum tree depth is set to 10, given the dataset's complexity and diverse features, as a deeper tree is required to uncover hidden patterns. The minimum sample split is set as 10, and the minimum sample leaf is set as 5. These settings enhance the algorithm's efficiency, while the minimum sample leaf ensures the representation of distinct patterns.

1.2.2 COMPARE

The performance and speed of the implemented decision tree regressor model will be compared with that of the sklearn decision tree regressor model.

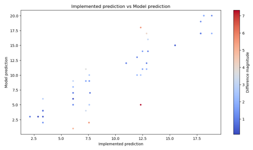

Figure 11: sklearn model vs implemented prediction

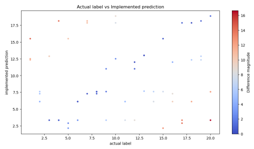

Figure 12: actual label vs implemented prediction

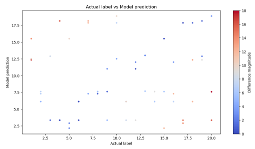

Figure 13: actual label vs sklearn model prediction

Regarding the running time, it takes 0.8963 to train the implemented model, while the sklearn model requires only 0.0101 seconds for training. However, the time taken to run the program is quite similar, with the implemented model using $5.016e-05$ seconds and sklearn model using $5.001e-05$ seconds to run the algorithm.

In Figure 12, when comparing the actual labels with the predictions from both the implemented model and the SKLearn model, it is observed that the data points exhibit considerable scatter without any clear pattern. A similar trend can also be seen in Figure 13, where the actual labels are compared with the model predictions. Since we are using a dataset with 50 features and 200 items for each feature, it is suspected that the hyperconfiguration may contribute to the inaccurate result. Specifically, we have set the minimum sample split to 10 and the minimum sample leaf to 5, which may not be optimal for splitting data into smaller parts with more nodes. As a result, the separation of data is not distinct and significant, leading to inaccurate predictions. The input data may also not be sufficiently informative, which likely contributes to the low accuracy of the model predictions and results underfitting.

1.2.3 IMPROVE

Several methods will be implemented to improve the model's accuracy. Figure 5 shows that image data has low diversity, and the concentrated data may lead to inaccurate results. Therefore, the image data will first be normalized using a MinMaxScaler. Additionally, as discussed in the data exploration section, the current dataset may be poorly organized, which results in insignificant outcomes. To address this, the data will be standardized to ensure consistent scaling.

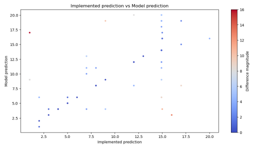

Figure 14: sklearn model vs implemented prediction

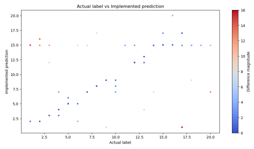

Figure 15: actual label vs implemented prediction

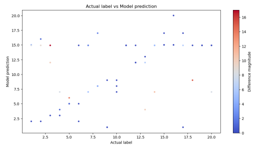

Figure 16: actual label vs sklearn model prediction

The model's accuracy has improved, as the actual labels now align closely with the model's predictions in Figure 15 and Figure 16, forming a near-linear relationship. It is believed that the significant improvement is attributed to the changes in the hyperparameters. By reducing the minimum sample split and leaf to 2, even pairs of insignificant data can now be distinguished and allocated to different tree nodes. Additionally, the increase in the maximum depth of the tree to 20 further contributes to this improvement. Also, normalizing and standardizing brain feature data has helped to create a normal distribution for the data, and the inclusion of the three datasets ensures the correctness and consistency of the data used.

1.3 Result Analysis and Visualization

1.3.1 PERFORMANCE

Table 1: Comparison of performance for Different Models

Model	Accuracy (ACC)	Precision	Recall	F1 Score
Scikit-learn Model	0.0333	0.0208	0.0333	0.0254
Implemented Model	0.1333	0.0685	0.1333	0.0890
Improved Model	0.3167	0.3075	0.3167	0.2866

While Decision Tree Regression has the disadvantages of being sensitive to outliers and noise and being prone to overfitting, the results still hold reference value for evaluating the algorithm's performance. When comparing the SKLearn model with the implemented model, it is found that the accuracy has improved by 300% However, it is clear that the accuracy is still relatively low, even with the use of the implemented model, due to the poor hyper configuration of the algorithm. As shown in Table 1, it can be seen that the performance of the new model has been improved drastically by 137%. With the tree nodes being split into different sub-sections, data can be distinguished more concisely, resulting in higher accuracy in predicting the outcome and labels. The new methodology of utilizing data is also a key factor in achieving the successful predictions.

1.3.2 VISUALIZATION

To visualize the performance of the implemented model, a confusion matrix is used to assess the accuracy of predictions and to identify the correct and incorrect classifications.

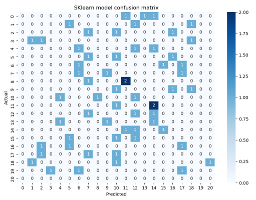

Figure 17: Sklearn model

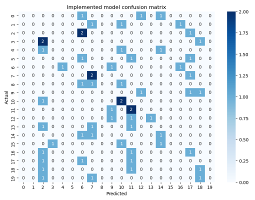

Figure 18: Implemented model

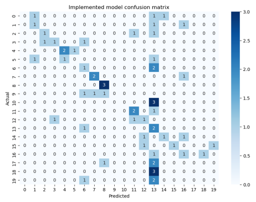

Figure 19: Improved model

1.3.3 ABLATION

Applying machine learning in real life can be challenging, especially when the dataset is complex and multi-dimensional. Therefore, the quality and distribution of datasets are crucial for generating accurate predictions. The potential issue discussed in section 1.1 may lead to inaccurate or meaningless predictions, as it contains outliers and insignificant data points that hinder the identification of hidden patterns. As a result, low accuracy prediction is generated, exemplifying the concept of "garbage in, garbage out." Decision tree regression can be particularly useful for handling complex and similar data by allocating it into nodes. Thus, it is evident that better-quality, generalized data is essential for effective regression methods, particularly decision tree regression, as features are categorized into different child nodes based on their distinct characteristics.

From the performance and visualization, the proposed model has shown an improvement in accuracy. The confusion matrix of Figure 17,Figure 18, Figure 19 reveals that both scikit-learn model and implemented model generally produce low accuracy results, with the actual and predicted outcomes not closely aligned. However, the implemented model demonstrates stronger performance, reflecting a higher level of accuracy in its predictions compared to the original implementation. While it may not be the optimal solution, as noise is still present and scattered across the graph, the accuracy has significantly increased, leading to more meaningful results from the machine learning model. In real life, achieving 100% accuracy in predicting cutting-edge problems is always challenging. Nonetheless, the implemented model provides a solution to issues such as uneven data distribution, outliers, and redundant data.

1.4 Paradigm Design and Data Splitting

1.4.1 PARADIGM

There are various methods for splitting a dataset into a training set and a testing set, such as 90%/10%, 80%/20%, and so on. It is important to note that each splitting method can lead to significantly different predicted results when applied to different machine learning models. Using a 85% training split helps avoid underfitting for complex models, while the remaining 15% serves as reliable data for evaluating the model's performance.

The dataset we are using consists of three sub-datasets: brain, image, and text features. Due to the complexity and insufficiency of the data, this highlights the importance of the new methodology we are implementing. We combine the features in brain, image, and text data, to form a dataset that consists of 60 features with 200 data items each. For the first testing, we will split the dataset into 180 data items for training and 20 data items for testing, and splitting it into 170 data items for training and 30 data items for testing, etc. In terms of scalability, models trained on larger datasets tend to generalize better, making them more adaptable to future scenarios or expanded datasets. This approach can significantly increase accuracy and is applicable across various industries to address cutting-edge problems in real life.

1.4.2 ADJUSTMENT

By using the new dataset split, the accuracy of the data prediction is expected to improve, resulting in a more generalized model. However, a few hyperparameters need to be adjusted to optimize the machine-learning model further. Firstly, the maximum depth is increased to 30, as this deepens the level of the tree when evaluating a complex, multidimensional dataset. Next, the minimum leaf sample is set to 1, meaning that at least 1 sample is required to create a leaf node. This ensures that the feature details remain distinct while splitting the data into reasonable sizes. Finally, the minimum sample split is set to 1, allowing for a deeper tree that can capture more intricate patterns, ultimately leading to more accurate results. It is expected that this newly improved model will show significantly higher accuracy compared to the previous one, with optimized performance from the updated implementation.

1.4.3 REFLECTION

Table 2: Comparison of performance for Different Models

Model	Accuracy (ACC)	Precision	Recall	F1 Score
Adjusted model	0.3167	0.3075	0.3167	0.2866
180 - 20 model	0.25	0.39	0.25	0.2625
170 - 30 Model	0.325	0.371875	0.325	0.3081
160 - 40 Model	0.24	0.2715	0.24	0.2212

From the new hyper-parameter tuning, it is observed that the split for 180-20 model has a much lower accuracy than the adjusted model, the accuracy has decreased 21.061% after implementing. While allocating more data to the training set may result in a well-rounded model with more cases, it may also have drawbacks due to the insufficient number of testing cases. From the result, it is clear that 10 testing cases are not enough to make accurate predictions. However, the 170-30 split model demonstrates higher accuracy compared to the improved model. It has increased by 2.62% in accuracy, 20.93% for precision, 2.62% for recall, and 7.50% for the F1 score. It is suggested that this may be the optimal split for the validation setting, providing an appropriate balance between testing and training cases. Data splitting plays a crucial role in improving the performance of the algorithms, as evidenced by the results. However, finding the optimal splitting method in real-world scenarios can be difficult and complex, as cutting-edge problems often involves multi-dimensional, complex datasets. It is crystal clear that regression methods play a critial role in simplifying problems, leading to more accurate predictions with the help of data splitting technique.

References

[1] Czajkowski, M. and Kretowski, M. (2016). The role of decision tree representation in regression problems – An evolutionary perspective. Applied Soft Computing, 48, pp.458–475.https://doi.org/10.1016/j.asoc.2016.07.007.

[2] Yang Long. (2025). Machine Learning Algorithms Lecture Slides. Retrieved from `https://blackboard.durham.ac.uk/ultra/courses/_57246_1/outline/file/_2472054_1` (Accessed: 20 January 2025).