Electron Collision Mass Prediction

P:30 Satyajeet Patil, Varad Patwardhan, Varad Sawant, Shubham Vasudeo Desai
Department of Computer Science, North Carolina State University
Raleigh, NC 27695
{smpatil2,vspatwar,vsawant,sdesai8}@ncsu.edu
Git-Hub-`https://github.ncsu.edu/vsawant/ALDA_Project_P30`

1 Background & Introduction

Studying electron collisions is instrumental in unraveling the secrets of the universe, as electrons are fundamental particles within the Standard Model of particle physics. These collisions serve as windows into the fundamental forces and particles that compose our world, potentially unearthing new particles or interactions that expand our comprehension of matter's basic constituents.

Furthermore, the ability to predict the outcomes of these collisions, especially in terms of electron mass, serves as a powerful tool for validating and testing theoretical models like the Standard Model. When experimental results align with these theoretical predictions, it strengthens our confidence in the accuracy of the model. Conversely, any disparities between observations and expectations may signal the need to explore new physics and broaden our existing understanding. Hence, harnessing machine learning techniques holds great promise in enhancing the precision and effectiveness of these predictions, ultimately advancing our grasp of particle physics

Problem Statement: Prediction of invariant mass of an electron formed by electron collisions. The dataset used is published by CERN (the European Organization for Nuclear Research). This dataset comprises 100,000 dielectron events within the invariant mass range of 2-110 GeV, intended for educational and outreach purposes.
The dataset: It includes the following columns:

- Run: The serial number of the run.
- Event: The serial number of the event.
- E1, E2: The total energy of the electron (in GeV) for electrons 1 and 2.
- px1,py1,pz1,px2,py2,pz2: The components of the momemtum of the electron 1 and 2 (GeV).
- pt1, pt2: The transverse momentum of the electron 1 and 2 (GeV).
- eta1, eta2: The pseudorapidity of the electron 1 and 2.
- phi1, phi2: The phi angle of the electron 1 and 2 (rad).
- Q1, Q2: The charge of the electron 1 and 2.
- M: The invariant mass of two electrons (in GeV).

To tackle this problem we have decided to use 4 machine learning algorithms mainly -

- **Decision Trees[1]:** Decision trees are important components of ensemble learning methods and have been widely used in a variety of scientific disciplines. Even though Decision Trees offer a clear view of the underlying decision-making process, overfitting is frequently mitigated by rigorous pruning and ensemble approaches when using Decision Trees in high-dimensional datasets.

- **Random Forest[2]:** Random Forest, an ensemble learning method that builds several decision trees during training, is a popular choice for dealing with large-scale and high-dimensional datasets. Its ability to prevent overfitting and improve generalization has made it a viable alternative for overcoming the limitations of conventional decision trees. One reason Random Forest is so popular is that it can capture intricate feature relationships while reducing the variation that comes with individual decision trees.

- **LightGBM[4]:** LightGBM, a high-performance gradient boosting framework, has gained attention in the field of data analytics due to its efficiency in handling huge datasets and capacity to handle high-dimensional sparse data. It can also handle categorical information while limiting overfitting via features such as early halting and regularization.

- **CatBoost[3]:** CatBoost, is well-known for its ability to efficiently handle categorical features. Its strong performance with complicated datasets and its ability to support GPU and parallel learning have made it a popular option for high-dimensional data processing.

2 Method

- Data Pre-processing: The columns 'Run' and 'Event' were not important in our analysis, so we removed them.

- Data Visualization :

  - scatter plot :Various scatter plots were generated using the mass variable as a reference. These scatter plots depicted the relationships between mass and other variables, including linear momentum, pseudorapidity, energy, phi angle, momentum of x, momentum of y and momentum of z directions.
  - Correlation Heatmap :A correlation heatmap was created to assess the relationships between different variables in the dataset.

- Data Cleaning:

  - To clean the data, there is a need to check for any outliers, if any and then assess their impact on our dataset during the analysis phase. In the dataset, the number of outliers are quite low.
  - Additionally, missing data or null-values are to be replaced by the median value.

- Model Building: Machine learning models were constructed using the Scikit-Learn library. The dataset was divided into a training set, which comprised 70% of the data, and a testing set with the remaining 30%. Four different algorithms were employed for the analysis: decision tree, random forest, LGBM (Light Gradient Boosting Machine), and CatBoost. These algorithms were used to build predictive models and assess their performance based on the provided data.

- Performance Evaluation : For performance evaluation purpose we used mean squared error method and R-squared.

3 Experiment Setup

- Dataset Loading: The first step of the experiment involved loading the dataset, which contained information related to the two electrons in the CERN dataset and the invariant mass created. This data was essential for understanding the particle behavior and interactions within the collider.
- Dataset Preprocessing: After loading the dataset, we checked for null values within the dataset. In the case of any missing values, we replaced these null values with the corresponding median values. This is to remove any bias and making sure the dataset is complete.
- Dataset Visualization: After addressing the missing values, we used Seaborn library to gain a comprehensive understanding of the dataset's underlying patterns and distributions. By creating various plots such as scatter plots, and heatmaps, we gained valuable insights into the relationships between different features, allowing us to identify potential trends and correlations within the dataset.
- Principal Component Analysis (PCA): To reduce the dimensionality of the dataset and highlight the most important features, we applied Principal Component Analysis (PCA). This helped us to transform the original set of correlated variables into a new set of linearly uncorrelated variables. By selecting the best components, we aimed to capture the maximum variance in the data, leading to a more efficient and effective training process for the subsequent models.
- Model Training: We proceeded to train various regression models, namely Decision Trees, Random Forest Regression, LightGBM, and CatBoost. These models were chosen for their ability to handle complex data and exhibit robust performance in predicting the invariant mass of two electrons in the CERN dataset. We used Standard Scalar to normalize the data to reduce any bias. For training and testing we used 70:30 split of data.
- Performance Evaluation: To assess each model, we calculated the R-squared (R2) value and Mean Squared Error (MSE) for each trained model. The R2 value provided insights into the variance in the dependent variable that could be explained by the independent variables, while the MSE quantified the average squared difference between the predicted and actual values. These metrics served as crucial indicators of the models' accuracy and generalization capabilities, enabling us to make accurate predictions of the invariant mass of two electrons in the CERN dataset.

4 Results

R-squared and MSE were used for performance evaluation of the predictive models

- R-squared : R-squared, also denoted as R^2 or the coefficient of determination, is a statistical metric employed in regression analysis for evaluating how well a regression model fits the data. It quantifies the portion of the variability in the dependent variable (the one we're trying to predict) that can be accounted for by the independent variables (those used for prediction) within the model.

- MSE : MSE, or Mean Squared Error, is a statistical metric used to assess the accuracy of predictive models. It measures the average squared difference between predicted and actual values in a dataset.
- Following observations were made:

Model Name	Mean Squared Error	R-2 Score
Decision Tree	75.921	0.879
Random Forest	27.016	0.956
LightGBM	24.827	0.960
CatBoost	5.605	0.993

Table 1: Results

5 Conclusion

Among all the models considered in this analysis, our inference is that CatBoost outperforms the other three models in both metrics, Mean Squared Error (MSE) and R-squared (R^2) Score. LightGBM and Random Forests exhibit comparable performance, while the Decision Tree model provides the highest level of accuracy. Looking ahead, an alternative approach to consider involves the utilization of Neural Networks for predicting our desired output.

References

[1] L. Breiman et al. "Classification and Regression Trees". In: *Biometrics* 40 (1984), p. 874. URL: https://api.semanticscholar.org/CorpusID:29458883.

[2] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[3] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. "CatBoost: gradient boosting with categorical features support". In: *CoRR* abs/1810.11363 (2018). arXiv: 1810.11363. URL: http://arxiv.org/abs/1810.11363.

[4] Robert P. Sheridan, Andy Liaw, and Matthew Tudor. *Light Gradient Boosting Machine as a Regression Method for Quantitative Structure-Activity Relationships*. 2021. arXiv: 2105.08626 [q-bio.BM].