A Machine Learning Approach to Predict the Invariant Mass of Dielectrons

Deepesh Narayan Raul, SK Somaiya

**Abstract**

The elementary particles of the universe and its interactions is said to be concise by the Standard Model of high energy physics or particle physics. Particle collider experiments has the capability to produce enormous and information-rich samples of data. Machine learning techniques can be used to develop how these data samples are interpreted, greatly expanding the discovery of potential present and future experiments. In this paper, data provided by Compact Muon Solenoid built on the Large Hadron Collider at CERN is analyzed to predict the invariant mass of two electrons using a statistical approach of machine learning.

***Keywords:*** *Machine Learning, CERN, Invariant Mass, Electrons, Regression, Feature Engineering.*

**Introduction**

Mid-1970s was when the Standard Model of particle physics was finalized upon the confirmation of quarks, since then, the evidence of top quark in 1995, tau neutrino in 2000 and the recent Higgs boson in 2012 have solidified the credence of the Standard Model. This theory describes that every observable objects within this universe is made from basic blocks called elementary particles, ruled by the four forces.

The invariant mass also known as rest mass or intrinsic mass is the fraction of the total mass of an object or system of objects that is independent of the total motion of the system. In particle physics,

Or where , in natural units,

A collider experiment is used in particle physics research by colliding pair of particles at very high kinetic energy. The Conseil Europ ́een pour la Recherche Nucl ́eaire or as we call it, CERN houses the world’s largest and highest energy particle collider, the Large Hadron Collider (LHS) and the Compact Muon Solenoid (CMS) a particle physics detectors. The CMS can generate huge amount of data for particle collisions at 0.9 – 13 TeV. The invariant mass is calculated using a different equation in a particle collider experiments (if particles are highly relativistic, or massless),

Where is defined as the angular position of particle in terms of azimuthal angle and pseudo rapidity . is the observed transverse momentum.

Using the machine learning we can use robust computing system alongside modern algorithms to observe and analyze insights from enormous amount of data quickly and efficiently. In this paper, we are not only trying to demonstrate the statistical significance of machine learning models in the field of particle physics by predicting the invariant mass of dielectrons based on the observation from the CMS detectors but also manipulating the features of the dataset by creating new features such that these features show higher relationship with the target variable, Invariant mass and increase the performance of the ML models.

**Data Collection and Processing**

The dataset used for this research is provided courtesy of the CERN open data portal. “Events with 2 electrons from 2010”, McCauley, Thomas, this dataset contains observations of 100K dielectrons events in the invariant mass of 2-110 GeV captured by the Compact Muon Solenoid. This data is organized in a CSV spreadsheet file and include the following observations collected by the CMS:

* **Run:** The run number of the event.
* **Event:** Number of each event
* **E1 and E2:** Total energy of the 2 electrons in GeV
* **px1, py1, pz1, px2, py2, and pz2:** Components of the momentum of the electrons in GeV
* **pt1 and pt2:** Transverse momentum of the electrons in GeV
* **phi1 and phi2:** phi angle of the electrons 1 and 2 in rad
* **eta1 and eta2:** The 2 electrons pseudo-rapidity
* **Q1 and Q2:** The charge of the electrons
* **Invariant Mass M**: The invariant mass of the dielectrons in GeV

As this dataset comes directly from the CERN open portal, it ensured that the observations are reliable, accurate and has been peer reviewed to be scientifically correct. The data released have been thoroughly analyzed and verified its accuracy through simulation events. Any results provided by this paper can be ensured to be true and accurate.

The dataset is further being processed by examining, cleaning, and analyzing the data and its features. First steps taken in data processing is the removal of duplicate data. As duplicate data are an extreme case of nonrandom sampling, as well as they bias any of the fitted models, leading to overfitting problems. In the cases for the CERNs dataset, these duplicates are not real data nor is intentionally oversampled. After analyses of the dataset, the target variable M i.e., the Invariant Mass have no values. As this research deals in predicting the values, the records are removed from the dataset instead of imputing the data which may lead to false results.

**Feature Engineering and Data Analysis**

This section provides the explanation of what manipulation of the features of the data was done and show the analysis of these new features. In statistics, and thereby Data Science, correlation analysis is done to calculate the level of relation between one variable to another. In other words, it measures the linear association between 2 variables.

* Where, is the Pearson correlation coefficient between 2 features, and
* is the mean of feature
* is the mean of feature

The values of the correlation coefficient are limited to between +1 and -1. If the coefficient value is close to +1, the 2 variables are perfectly positive interrelated; that is if one variable increase positively, the other also increases. On the other hand, a coefficient value that tends close towards -1, the 2 variables are perfectly negative interrelated: if one variable increases, the other decreases perfectly in the opposite direction. With a coefficient value near 0 there isn’t any interrelation. For the research purpose we are looking for correlation coefficient that tends towards +1 and -1 between each independent features and the target M.

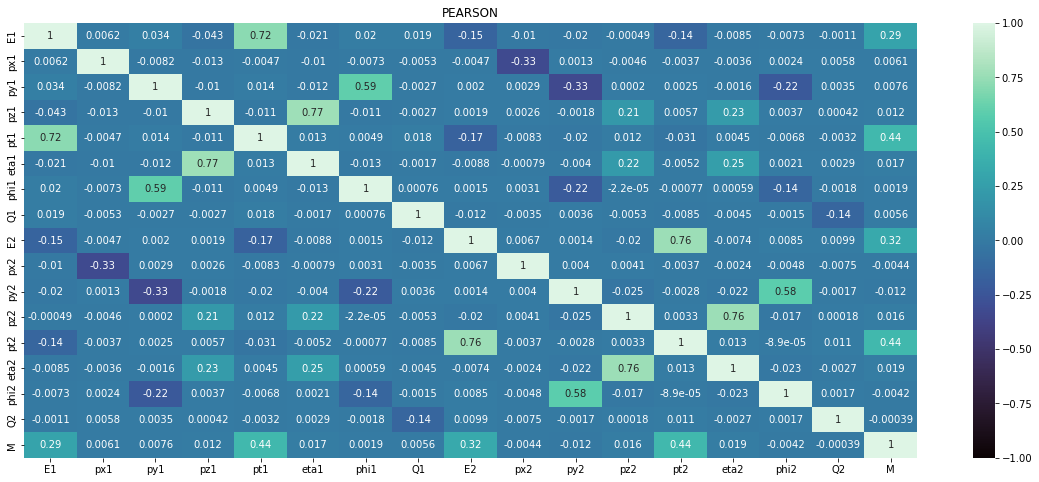


Figure Heatmap for the correlation coefficients

Figure 1 is the heatmap for the Pearson’s correlation coefficient values of the dataset shows us the values between each feature. The major focus here is the coefficient value between the target value M and each independent variable. The heatmap shows that E1, E2, pt1 and pt2 features have high correlation values 0.29, 0.32, 0.44 and 0.44 respectively, with Invariant Mass compared to all the other features.

This research presents new features that is created using the original features.

* **E12:** the product of features E1 and E2.
* **pt12:** the product of features pt1 and pt2
* **Similarly, eta12, px12, py12, pz12, phi12,** the products for their respective features.

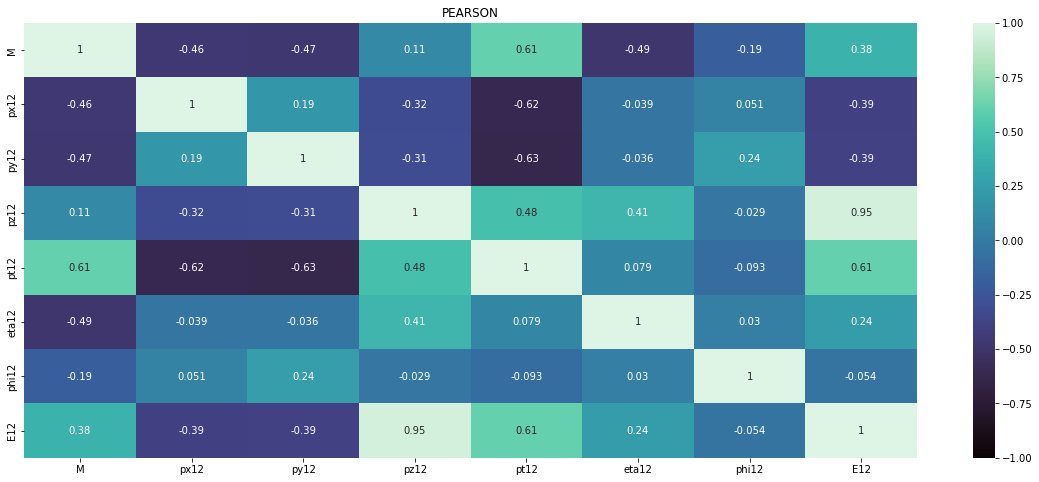


Figure Correlation Heatmap for new features

The heatmap from Figure 2 shows the correlation between the new features with the dependent variable M. With the min correlation value to be 0.11, we can say that that the new features have good relation to the dependent variable M

|  |  |  |  |
| --- | --- | --- | --- |
| Features | \_1 | \_2 | \_12 |
| px\_ | 0.0061 | -0.0044 | -0.46 |
| py\_ | 0.0076 | -0.012 | -0.47 |
| pz\_ | 0.012 | 0.016 | 0.11 |
| pt\_ | 0.44 | 0.44 | 0.61 |
| eta\_ | 0.017 | 0.019 | -0.49 |
| phi\_ | 0.0019 | -0.0042 | -0.19 |
| E\_ | 0.29 | 0.32 | 0.38 |

Table Correlation coefficient value comparisons

The table 2 shows the comparison of the correlation coefficient values between the original two features and the new features created using them. We can infer that the new features highly correlated to M compared to the parent features.

With the introduction of new features to the dataset, 2 test needs to be done. First, perform the p-value test, to prove the significance of the correlation that is evaluated and secondly deal with the problem of multicollinearity.

The p-values test is a measure of significance to validate a hypothesis against the observed results or data. Here, the null hypothesis is that there is no correlation between the features and the target. The threshold for this test is , wherein the null hypothesis is rejected. The p-values for all the features, including the newly created ones, are calculated using Ordinary Least Square Regression model. Every feature has a p-value less than 0.05 and therefore null hypothesis is rejected, proving the correlations found are significant to use.

As new features are created, the problem of multicollinearity needs to be solved, where 2 or more independent variables are highly correlated with each other. High multicollinearity causes inaccurate results of regression analysis, due to the unstable and biased estimation of the regression coefficients, increases standard error and the variance of the coefficients which in turn also decreases the statistical power. For this research purpose, the highly correlated independent variables are removed.

Multicollinearity is detected by calculating the variance inflation factor (VIF) of each feature in the dataset, it is the measure of how much the standard error of the estimate of the coefficient increased due to multicollinearity.

Here, denotes the multiple correlation coefficient between independent variable and remaining variables ()

The threshold for the VIF value for this research is less than 10. Above which the feature is removed and not used at all. For the original data all features (excluding M, the dependent variable) have VIF values below 10.

With the introduction of the new features to the dataset the VIF values are recalculated.

|  |  |
| --- | --- |
| Features | VIF Value |
| pt2 | 10.668826 |
| pz12 | 93.354042 |
| E12 | 118.840222 |

Table 3 Features with VIF values > 10

Table 3 reveals that 3 features have strong multicollinearity problems. As E12 and pt2 have high correlation with M compared to pz12, this feature is dropped and the new recalculated VIF values for all remaining features are within the threshold.

**Regression Analysis and Model Performance**

In the previous sections, the analysis of the relationship between the features and the target through which new features are introduced and chosen for the research. Regression Analysis is a statistical modelling technique used to estimate the relationships between the dependent variable(s) and the independent variables. Regression analysis is primarily used for two distinct purposes, firstly prediction and forecasting and second, to infer causal relationships between dependent variables and independent variables. The first purpose overlaps with machine learning field and the focus for the paper. The simplest and basic regression model is the linear model.

Where, is the dependent variable, is the model coefficient, is the intercept, is the coefficient for the , the nth feature.

The main goal of this model is to find the best fit line where the error between the actual values and the predicted values is minimized.

In the training process, the model coefficient that best describes the input variables are found. A cost function, also known as loss function, is also chosen for the model which helps measure the error so that the model can minimize it.

Unlike in a classification problem where the evaluation metric is generally chosen to be accuracy, Root Mean Squared Error, RMSE, is used to evaluate regression problems, as the values are continuous

Here, is the total number of observations, are the actual values and are the predicted values. A RMSE value needs to be minimized, a model with RMSE value closer to 0 is considered a good model.

For this paper, another metric is used with RMSE, R-squared score is a statistical measure that evaluates the goodness of fit of the model.

takes a value between 0 to 1, where a value closer to 1, the better the regression model fits the actual observations.

Ten regression models are selected and fitted to the original dataset and the new added feature dataset. Following are the regression models used for this research.

Decision Tree Regression is a regression model that uses the concept of decision tree. A model that uses a tree-like model of decisions to predict the target value.

Gradient Boosting Regression is a machine learning ensemble technique which predicts target output by combining several weak learners in particular decision tree.

LASSO stands for Least Absolute Shrinkage and Selection Operator is a regularization technique, specifically L1 regularization.

|  |  |  |
| --- | --- | --- |
| Model | R2 Score | RMSE |
| DecisionTreeRegressor | 0.66747 | 14.55148 |
| CatBoostRegressor | 0.99037 | 2.47511 |
| ElasticNet | 0.39832 | 19.56982 |
| GradientBoostingRegressor | 0.74486 | 12.74279 |
| Lasso | 0.39835 | 19.56938 |
| LGBMRegressor | 0.95843 | 5.14299 |
| PLSRegression | 0.39183 | 19.67583 |
| RandomForestRegressor | 0.90155 | 7.91384 |
| Ridge | 0.39804 | 19.57434 |
| XGBRegressor | 0.96420 | 4.77334 |

Table 4 Evaluation metrics using original dataset

|  |  |  |
| --- | --- | --- |
| Model | R2 Score | RMSE |
| DecisionTreeRegressor | 0.97820 | 3.72371 |
| CatBoostRegressor | 0.99699 | 1.38363 |
| ElasticNet | 0.81897 | 10.73765 |
| GradientBoostingRegressor | 0.98590 | 2.99343 |
| Lasso | 0.82125 | 10.66939 |
| LGBMRegressor | 0.99401 | 1.95138 |
| PLSRegression | 0.79007 | 11.56486 |
| RandomForestRegressor | 0.99092 | 2.40208 |
| Ridge | 0.82179 | 10.65290 |
| XGBRegressor | 0.99333 | 2.05787 |

Table 5 Evaluation metrics using dataset with new features

The lasso regression model uses shrinkage, where feature values are minimized towards the central point as mean. Very useful for large number of features as it automatically performs feature selection.

Ridge regression is a regression model that uses L2 regularization technique. Like Lasso, this model uses shrinkage, helpful to prevent multicollinearity and reduce model complexity through coefficient shrinkage.

Random Forest Regressor takes same or multiple algorithms, and a model is put together that’s more effective than the original. In this case, multiple decision trees are built. This technique is called ensemble learning.

PLS Regression also known as Partial Least Squares regression is a method that minimizes the predictors to a smaller set of uncorrelated components and performs least squares regression on these components. Unlike multiple regression, PLS does not presume the predictors fixed.

Elastic net linear regression is a regularized regression that overcomes the limitations of both the lasso and ridge techniques by linearly combining the L1 and L2 penalties of lasso and ridge models.

Decision Tree regression, Gradient Boost, Lasso, Ridge, Elastic net, Random Forest Regressor, and the PLS regression are provided by the scikit-learn library.

XGBoost also known as eXtreme Gradient Boosting is a regularizing gradient boosting framework for many languages such as C++, python, scala etc. It is a open-sourced, distributed, scalable gradient-boosted decision tree (GBDT) library. The trees are grown level-wise or row by row. Developed by The XGBoost Contributors.

LightGBM, or Light Gradient Boosting Machine is framework of gradient boosting based on decision trees. This type of framework helps reduces memory usage and increase the efficiency of the model. Supports many different algorithms such as GBDT, GBRT(Gradient Boosting Regression Trees), MART(Multiple Additive Regression Trees) and GBM. LGBM has a lot of XGBoost advantages, including early stopping, multiple loss function, sparse optimization, parallel training. But unlike XGBoost, the tree is grown leaf-wise. Developed by Microsoft and Guolin Ke.

CatBoost is also a gradient boosting framework, developed by Yandex. This library is open-source and models can be built in various languages, including C++, python, R and many more. Unlike similar gradient boosting models, CatBoost grows oblivious trees, these trees are grown by enforcing the rule that all nodes at the same level, test the same predictor with the same condition and therefore bitwise operations can be used to calculate the index of the leaf.

Tables 4 and 5 are the evaluation metrics of all the regression models that are fitted on the original dataset and the new dataset respectively. These metrics are calculated by taking the means of the R2 score and RMSE values evaluated across 10 – fold cross-validation. It is evident that the new features that were engineered have a positive impact on the performance across all the regression models. Keep in mind these models are fitted on their default hyperparameters. The four gradient boosting regression models, GBR, LGBM, XGBoost, and CatBoost have the best performance out off the other types of regression algorithms.

The CatBoost regression is the best model with a R2 score of 0.99699 and a RMSE value of 1.38363. This paper will be using the CatBoost model for the prediction of invariant mass from this point onwards as it has the best performance out of all the models.

**Hyperparameter Optimization and Normalization of Data**

In the previous section, it is mentioned that the models were evaluated on their default hyperparameters. Hyperparameters are those parameters that are used to manage the learning process of the machine learning model. These values are not derived by fitting the model to that the training set. These values mainly influence the speed and quality of the learning process. Hyperparameter tuning is the method of choosing a set of hyperparameters for a machine learning process. This done by finding the best hyperparameters, that minimizes a predefined loss function. For this research the loss function is the RMSE value.

There are three popular approaches for hyperparameter tuning:

1. Grid Search
2. Random Search
3. Bayesian Optimization

Grid Search or parameter sweep is an exhaustive search of the hyperparameter, in simpler terms, every possible combination of hyperparameters, in a grid, are tested and best one is selected. As this goes through every combination it uses a lot of resources.

Random Search is an offset of Grid Search where the exhaustive enumeration of all combinations is replaced by selecting these parameters randomly and testing these selected values to find the best hypermeters. This cuts down the time to find the parameters but may not necessarily return the best parameters.

Both these methods have several disadvantages, Grid search takes a long time and a lot of resources, and random search may cut down the time to find the parameter, the values found may not be the best parameters. These two methods also have another significant issue, these algorithms are completely uniformed by the previous evaluations, and therefore spend substantial amount of time to evaluate hyperparameters that may not be the best.

Bayesian optimization approach, differs from grid search and random search as it keeps a track of previous evaluations