MACHINE LEARNING PROJECT

REGRESSION ANALYSIS REPORT

REPORT ON THE ANALYSIS OF DIFFERENT REGRESSION MODELS

SHIVANSH VERMA

11.06.2023

2nd YEAR

UNIVERSITY SCHOOL OF AUTOMATION AND ROBOTICS, GGSIPU EDC

ABSTRACT

In this report, I have done research on various regression models to identify the best one among them. I have performed six regression models on the available air quality dataset, starting from Multiple Linear Regression to Support Vector Regression. Regression analysis is the study of modeling the relationship between dependent variable and independent variable. I had to clean the data before using it in any regression analysis as the data comes from various sources. The data also had to be normalized as to not overpower any attribute in the determination of the target attribute. I used different libraries of sklearn, pandas, numpy and matplotlib to import different regression models, plots, and some table information for my research. In conclusion, I found out that Random Forest Regressor is the best regression model amongst the other models that I used in this research i.e., Multiple Linear Regression, Lasso Regression, Ridge Regression, Decision Tree Regression and Support Vector Regression.

KEY WORDS

Regression, Data preprocessing, Random Forest Regression, Decision Tree Regression, scikit-learn.

INTRODUCTION

Regression analysis is the field or study of plotting or modeling a relationship between dependent and independent variables. I have used six different regression models in this research: Multiple Linear Regression, Lasso Regression, Ridge Regression, Decision Tree Regression, Random Forest Regression and Support Vector Regression. We will talk about them in the next subtopic.

PROPOSED METHODOLOGY

1. **Data Set:** The dataset contains 9358 instances of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multi

Sensor Device. The device was located on the field in a significantly polluted area, at road level,within an Italian city. Data were recorded from March 2004 to February 2005 (one year)representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Ground Truth hourly averaged concentrations for CO, Non Methane Hydrocarbons, Benzene, Total Nitrogen Oxides (NOx) and Nitrogen Dioxide (NO2) and were provided by a co-located reference certified analyzer. Our dataset has 15 attributes as follows:

0 Date (DD/MM/YYYY)\

1 Time (HH.MM.SS)

2 True hourly averaged concentration CO in mg/m^3 (reference analyzer)

3 PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)

4 True hourly averaged overall Non Methane HydroCarbons concentration in micro g/m^3 (reference analyzer)

5 True hourly averaged Benzene concentration in microg/m^3 (reference analyzer)

6 PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)

7 True hourly averaged NOx concentration in ppb (reference analyzer)

8 PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)

9 True hourly averaged NO2 concentration in microg/m^3 (reference analyzer)

10 PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
11 PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)

12 Temperature in Â°C

13 Relative Humidity (%)

14 AH Absolute Humidity

2. **Pre processing:** We had to do some preprocessing as our data was raw and was not cleaned. We dropped the unnecessary rows that had null values as the null values present in the dataset was less than 5% and columns to clean our data and we normalized the data too so as to not overpower any attribute in the determination of dependent or target attribute. We used the ‘dropna’ and ‘drop’ commands of pandas to drop the rows and columns respectively. We used a ‘StandardScaler’ sklearn module to

standardize or normalize the data.

3. **Multiple Linear Regression:** The dependent variable is continuous in nature and the model shows a linear relationship between them. It has one dependent variable and multiple independent variables. The equation used for this model is:

   $y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \ldots + \alpha_m x_m$

   α_i = Regression coefficient

   x_i = Independent variable

   y = Dependent variable

   The more the value of alpha, the more is the importance of the independent variable, more is its effect.

4. **Ridge Regression:** It is the technique used to reduce the complexity of the generated model. It is done to reduce the computational expenses. It is done by reducing the scaling down the value of coefficient of regression. The equation used for this model is as follows:

   Ridge Regression = Loss + $\alpha||W||^2$

   Loss = The difference between the actual and predicted value.

   $||W||^2 = W_1^2 + W_2^2 + W_3^2 + \ldots + W_n^2$

   By increasing the value of α, the value of coefficient of regression decreases, which is ultimate motive of ridge regression.

5. **Lasso Regression:** It is the technique used to reduce the complexity of the generated model. It is done to reduce the computational expenses. It is done by reducing the scaling down the value of coefficient of regression. The equation used for this model is as follows:

   Ridge Regression = Loss + $\alpha||W||$

   Loss = The difference between the actual and predicted value.

   $||W|| = W_1 + W_2 + W_3 + \ldots + W_n$

   By increasing the value of α, the value of coefficient can be zero unlike in Ridge regression. Lasso Regression also acts as feature selection.

6. **Support Vector Regression:** Support Vector Regressor (SVR) is a regression algorithm based on Support Vector Machines (SVMs). It minimizes errors within a specified margin around predicted values, making it suitable for non-linear relationships. SVR uses support vectors, a kernel function, and a regularization parameter (C) to define the model. It employs a loss function to penalize errors beyond the margin. SVR is evaluated using metrics like MSE and R-squared. The equation used for this model is as follows:

   y = w^T * x + b

   y is the predicted output (regression target).

   w represents the weight vector.

   x is the input features.

   b is the bias term.

   The objective of SVR is to find the optimal values for w and b that minimize the error within a specified margin or tube around the predicted values, taking into account the regularization parameter C and the chosen kernel function. The specific form of the equation and the optimization problem can vary depending on the chosen kernel and the formulation of the SVR algorithm.

7. **Decision Tree Regressor:** Decision Tree Regressor is a tree-based algorithm used for regression. It recursively partitions the input space, making predictions based on target values within each region. It handles non-linear relationships and offers interpretability. Overfitting can be mitigated through pruning or ensemble methods. Evaluation is done using metrics like MSE or R-squared.

   (i) Starting at the root node, the algorithm evaluates a specific feature and its threshold.

   (ii) If the condition is satisfied (feature value <= threshold), the algorithm follows the left branch. Otherwise, it follows the right branch.

   (iii) This process continues recursively at each subsequent node until reaching a leaf node.

   (iv) The predicted value at the leaf node is typically the average or majority value of the target variable within that leaf.

8. **Random Forest Regressor:** Random Forest Regressor combines decision trees in an ensemble for regression tasks. It uses bootstrap sampling and features randomness to

improve predictive accuracy and generalization. While interpretability is reduced, feature importance can be assessed. Hyperparameters can be tuned, and evaluation is done using metrics like MSE or R-squared.

Steps in Random Forest Regressor:

1. Randomly sample subsets of the training data using bootstrap sampling.

2. Build multiple decision trees, each using a random subset of features.

3. Train each decision tree by optimizing splits to minimize error (e.g., MSE).

4. Aggregate predictions from all trees, typically by averaging, for the final prediction.

5. Assess feature importance by measuring the impact of random feature permutation.

6. Tune hyperparameters like number of trees, tree depth, and feature subset size.

7. Make predictions on new data by combining outputs from individual trees.

RESULT AND DISCUSSIONS

By applying different models on the obtained dataset, we found out the following results. We trained our data from the dataset given to us and we tested our model by giving it test data to find out the prediction accuracy. We used an R-squared score to check the fitness of our model. The R-squared (R2) score, also known as the coefficient of determination, is a statistical measure used to evaluate the goodness of fit of a regression model. It provides an indication of how well the predicted values from the model align with the actual observed values of the target variable. The following results were obtained for different models on our test data:

1. Multiple Linear Regression: R-squared (R2) score: 0.9993618893773653

2. Ridge Regression: R-squared (R2) score: 0.9991702740871099

3. Lasso Regression: R-squared (R2) score: 0.9218822435757943

4. Support Vector Regression: R-squared (R2) score: 0.9193105132565409

5. Decision Tree Regression: R-squared (R2) score: 0.9999987285556645

6. Random Forest Regression: R-squared (R2) score: 0.9999998323105906

CONCLUSION AND FUTURE WORK

By applying different models on the obtained dataset, we come to the conclusion that Random Forest Regression is the best model that can be used to predict the test data. The best regression technique found in this research was Random Forest Regression and the second best was Decision Tree Regression followed by the others. We can say that if anyone wants to apply a regression analysis on a given dataset they should use Random Forest Regression to get the best or optimal or closely predicted values to the test data.

REFERENCES

1. **For Dataset:** Vito,Saverio. (2016). Air Quality. UCI Machine Learning Repository. https://doi.org/10.24432/C59K5F.
2. For Models: year={2011}

   }

3. **@article**{scikit-learn,

    title={Scikit-learn: Machine Learning **in** {P}ython},

    author={Pedregosa, F. **and** Varoquaux, G. **and** Gramfort, A. **and** Michel, V.

   and Thirion, B. **and** Grisel, O. **and** Blondel, M. **and** Prettenhofer, P.

   and Weiss, R. **and** Dubourg, V. **and** Vanderplas, J. **and** Passos, A. **and**

   Cournapeau, D. **and** Brucher, M. **and** Perrot, M. **and** Duchesnay, E.},

```
    journal={Journal of Machine Learning Research},
    volume={12},
    pages={2825--2830},
```

4. **For information:** Wikipedia